

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-167028

(43)Date of publication of application : 25.06.1996

(51)Int.Cl.

G06T 7/00

(21)Application number : 06-308581

(71)Applicant : TOPPAN PRINTING CO LTD

(22)Date of filing : 13.12.1994

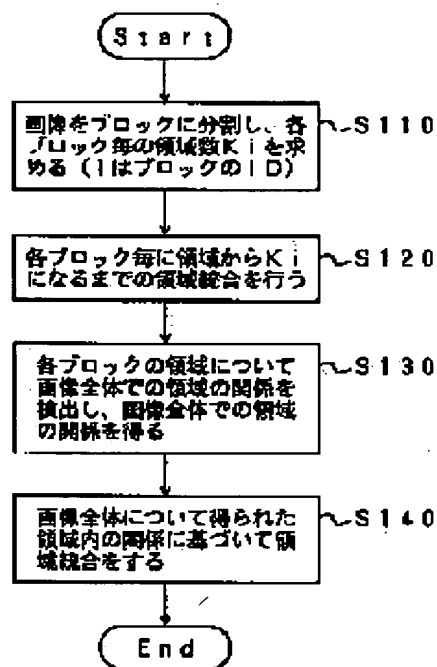
(72)Inventor : ATOZAWA NAOHITO

(54) IMAGE PROCESSING METHOD

(57)Abstract:

PURPOSE: To perform processing by a computer with small storage capacity by dividing an image into necessary units and integrates areas as to their constituent pixels, putting images of respective units together into one screen image as an initial cluster, and coupling areas as to the cluster.

CONSTITUTION: One image is divided into divided areas of easy-to-process size and the number of pixels to be processed is decreased to facilitate the handling (S110). Then the individual divided images are hierarchically clustered (S120). Then all the divided images are put together and handled as one image, i.e., the initial cluster, which is hierarchically clustered (S130 and S140). At this time, the history of the integration is saved together with color component information on the current cluster. The history of integration is traced back to know and extract a cluster having information on a desired color component, thereby dividing a desired area part from the image.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

JP-A-H08-167028

Image Processing Method

[What is claimed is:]

- 5 1. An image processing method, comprising:
 a first area integration processing step of
dividing an image to be processed into necessary units
and integrating areas as to their constituent pixels
for each divided image; and
10 a second area integration processing step of
putting images of respective units together into one
screen images as an initial cluster, integrating areas
as to the cluster and also storing its integration
history.
- 15 2. An image processing method, comprising:
 a first area integration processing step of
dividing an image to be processed into necessary units
and integrating areas as to their constituent pixels
for each divided image; and
20 a second area integration processing step of
putting images of respective units together into one
screen images as an initial cluster, integrating areas
as to the cluster and also storing its color component
information of the cluster to manage history information
25 of area integration by a binary tree.

3. The image processing method according to claim 1 or 2, wherein

area integration has adjacent relationship and a cluster with the minimum non-similarity degree in color space integrated with priority.

4. The image processing method according to claim 1 or 2, wherein

integration history of a cluster to be integrated including information indicating area integration order and similarity degree between integrated areas and coordinate information of a cluster to be integrated is stored and is managed by a binary tree.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-167028

(43) 公開日 平成8年(1996)6月25日

(51) IntCl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 T 7/00		9061-5H	G 0 6 F 15/ 70	3 3 0 Q

審査請求 未請求 請求項の数4 O L (全 31 頁)

(21) 出願番号 特願平6-308581

(22) 出願日 平成6年(1994)12月13日

(71) 出願人 000003193

凸版印刷株式会社

東京都台東区台東1丁目5番1号

(72) 発明者 後澤 尚人

東京都台東区台東一丁目5番1号 凸版印刷株式会社内

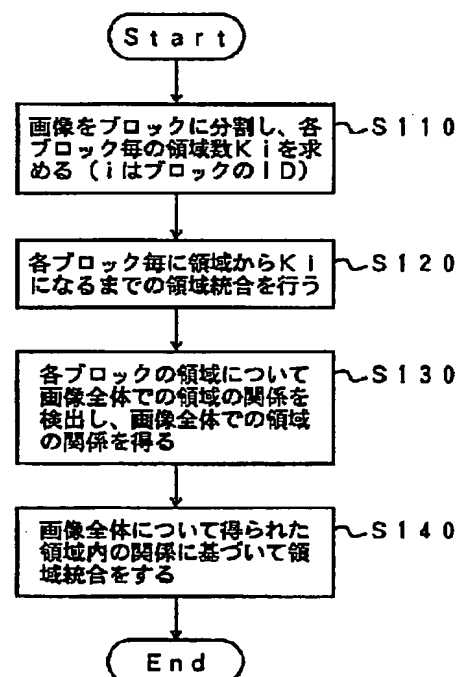
(74) 代理人 弁理士 鈴江 武彦

(54) 【発明の名称】 画像処理方法

(57) 【要約】

【目的】 処理に膨大なメモリ容量を要することなく画像の目的部分を抽出処理することができるようにする。

【構成】 処理対象の画像を区分けし、この区分けしたものを単位に、第1段階での領域統合を行い、この領域統合が済んだ後、この処理済みの画像を取り纏めて元の1画面分の画像として扱い、第2段階での領域統合を行って、目的の領域の抽出を可能にする。本発明では処理対象の画像を分割あるいはブロック化して領域統合処理を行うことで、領域統合処理毎の一度に取り扱う画素・クラスター・領域の個数を限定することができようにし、これによって記憶資源の小さな電子計算機においても演算処理が可能となるようにした。



【特許請求の範囲】

【請求項 1】 処理対象の画像を所要の単位に分け、この分けた画像を単位に、その構成画素について領域統合する第 1 の領域統合処理工程と、

この第 1 の領域統合処理工程により得られた上記各単位の画像を取り纏めてこれらを初期クラスタとする 1 画面分の画像とし、この画像のクラスタについて領域統合を行うとともに、統合履歴を保存する第 2 の領域統合処理工程とよりなることを特徴とする画像処理方法。

【請求項 2】 処理対象の画像を所要の単位に分け、この分けた画像を単位に、その構成画素について領域統合する第 1 の領域統合処理工程と、

この第 1 の領域統合処理工程により得られた上記各単位の画像を取り纏めてこれらを初期クラスタとする 1 画面分の画像とし、この画像のクラスタについて領域統合を行うとともに、領域統合の履歴情報を二分木により管理すべく情報保存する第 2 の領域統合処理工程とよりなることを特徴とする画像処理方法。

【請求項 3】 領域統合は隣接関係があり、色空間上の非類似度最小のクラスタから優先的に統合することを経験とする請求項 1 または 2 いずれか一方記載の画像処理方法。

【請求項 4】 領域統合の順序および領域統合された領域間の類似度を表わす情報、統合されるクラスタの座標情報を含む、統合されるクラスタの統合履歴を保存するとともに、二分木で管理することを特徴とする請求項 1 または 2 いずれか一方記載の画像処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、画像処理方法にかかわり、特に画像データにある特徴を有する領域毎に分割するための画像処理方法に関する。

【0002】

【従来の技術】 印刷・写真・絵画等の分野においては、画像の明度・色相・彩度・座標（位置）等を総合的にとらえ、「ライト、中間、シャドウ」のように画像の部分を分類することが多い。そして、これは例えば、人物を表す写真の校正段階において「人物の“ライト”をさらに明るく」等と云った具合に、画像のある部分を特定するためにこれらの「ライト、中間、シャドウ」という概念が用いられる。そして、「ライト、中間、シャドウ」の区分け判断は、専ら作業員個人の判断に委ねられており、この判断をするに当たって、作業員は画像の明度・色相・彩度・座標等を総合的に考慮することになる。

【0003】 ところで近年、印刷等の分野においては、原稿画像をコンピュータにより処理することが多くなった。そのため、「ライト、中間、シャドウ」の分類についても同様にコンピュータにより自動的に抽出処理することが好ましい。コンピュータにより画像中のある領域を抽出する場合に利用可能な一般的な手法としては、

“濃度差を有する画像を所定の閾値を用いて領域分割を行う方法”、“濃度または色度等のヒストグラムにより決定された閾値を用いて領域分割を行う方法”、“画像のエッジを領域の境界線と看做して領域分割を行う方法”等がある。

【0004】 しかしながら、これら従来の画像処理方法は、いずれも画像の明度、色相、彩度、座標等の各要素を総合的にとらえるものではなく、一つの要素についてのみを対象として判断し、領域分割を行うものである。故に、従来の画像処理方法により分割された領域は、

「ライト、中間、シャドウ」という概念による領域とは異なったものになってしまう。すなわち、上述した画像処理方法では画像を「ライト、中間、シャドウ」のような概念に当て嵌めて分割しようとしても、人間の認識に合うような領域分割はできない。

【0005】 また、「ライト、中間、シャドウ」の分割・分類に階層性があることに着目すれば、階層的クラスタ分析法を画像に適用して、この「ライト、中間、シャドウ」の各領域を分割・分類することも考えられる。ところが、この場合には以下の問題が生じる。

【0006】 それは階層的クラスタリング（階層的領域分け）を行うためには、全ての画素対についての色空間上の距離を計算し、この中から最小距離のクラスタ対を見つけ出さなければならないと云う点である。このため、画素数が増えると、計算対象とするクラスタ対の数は膨大な数となり、処理時間および必要とするメモリ容量は膨大なものになってしまう。

【0007】 例えば、画素数が 512×512 の画像データにあつては、全画素数 n が $n = 512 \times 512 = 262144$ になるが、この 262144 画素での画素対の組み合わせの数は、 $nC_2 = 3,436 \times 10^6$ にもなる。そして、このような膨大な数のデータを保持するには、およそ $200 \sim 400 \text{ GBytes}$ もの大容量メモリを必要とする。

【0008】 また、画素数 n についての画素対の数は $nC_2 = n(n+1)/2$ で表されることから、演算対象となる画素対の数は画素数 n の二乗に比例して増加することが確認できる。従って、かかる方法を利用してコンピュータで処理するには、あまりにも膨大な計算時間と膨大なメモリ容量を必要とすることとなるから、高精細画像を対象としたクラスタリングには適用できず、上述のような「ライト、中間、シャドウ」の各領域に分割する処理に適用するには不適當である。

【0009】 故に、従来の画像処理方法はいずれも画像を「ライト、中間、シャドウ」の各領域に分割する処理に適用するに不向きである。そこで、これら問題を解決するために本発明者は、特願平 5-340077 号（以下、先行技術と呼ぶ）に示す如き階層的な領域抽出方法を提案した。この方法を次に説明する。

【0010】 【先行技術】 すなわち、この先行技術に適

用した手法は、

【1】 画像データ中の対比される2つの領域の類似度を、色空間上における当該2つの領域の座標に基づき算出し、上記画像データ中の一の領域に隣接した複数の他の領域のうち、当該一の領域との類似度が最大となる他の領域を当該一の領域に順次統合する。

【0011】 【1】 また、上述の【1】の処理に加え、さらに領域統合の順序および領域統合された領域間の類似度を表わす履歴データを生成し、上記履歴データに従い領域統合された画像データのうち、所定範囲の類似度の領域対を、領域統合の順序と逆の順序で順次分割する。

【0012】 これらのうち【1】の方法は、まず画像データ中の対比される2つの領域（クラスタ）の類似度を、色空間上における当該2つの領域の座標に基づき算出する。この演算結果は、2つの領域の色度等が近似している場合に、類似度は大きくなる。そして、画像データ中の一の領域に隣接した複数の他の領域の中から、当該一の領域との類似度が最大となる他の領域を当該一の領域に順次統合する。その結果、色度の近似した領域同士が統合され、領域間のコントラストを十分に高くすることができることになる。

【0013】 また、【11】の方法では、まず、画像データ中の対比される2つの領域の類似度を、色空間上における当該2つの領域の座標に基づき算出する。そして、上記画像データ中の一の領域に隣接した複数の他の領域のうち、当該一の領域との類似度が最大となる他の領域を当該一の領域に順次統合する。このとき、領域統合の順序および領域統合された領域間の類似度を表わす履歴データを生成しておく。そして、領域統合された画像データのうち、上記履歴データに従い、所定範囲の類似度の領域対を、領域統合の順序と逆の順序で順次分割する。すなわち、領域統合された画像データは、ある程度特徴の近似した領域毎に再度分割されるので、画像データ中から所望の領域を抽出することができるようになる。

【0014】 さらに詳細を説明しておく。ここで先行技術において採用している階層的クラスタリングの概要を先ず説明する。先行技術における階層的クラスタリングとは、色相、明度、彩度等の特徴が近似している（つまり類似度の大きい）領域（クラスタ）同士を順次統合していくことにより画像の領域分割を行うと云う処理である。

【0015】 この類似度の大きい領域同士を順次統合していくと云う手法によれば、与えられた画像について人物と背景との領域に分割すると云ったことが可能となる。先行技術の手法を採用した画像処理装置（コンピュータ）に、例えば、物と背景からなる n 画素構成のカラー画像のデータ（画像データ）を与えたとする。するとこの画像処理装置では、 n 画素の画像データ中の各画素

の色空間上の座標点を初期クラスタとして背景部分と人物部分の切り分けをする処理を進めることになる。

【0016】 色空間上における n 個のクラスタは、背景を表す画素と、人物を表す画素とに対応して2つの固まった分布をなす。そこで、先行技術では、画面を構成する n 個の初期クラスタの中の一のクラスタに着目し、色空間上において当該クラスタの色度に近い他のクラスタを探し出し、統合するようにする。すなわち、画像処理装置は一のクラスタと他の全てのクラスタとの色度等を色空間上の座標により計算し、色度の近似した他のクラスタを探し出す。

【0017】 このようにして検出されたクラスタ対は、実画像における色相、明度、彩度等が互いに近似したものとなる。従って、この検出されたクラスタ対を統合する。この検出されたクラスタ対を統合することにより、クラスタの数は $n-1$ 個となる。この処理を繰り返し実行し、色度の近似したクラスタ同士を順次統合すると、クラスタの数は次第に減り、最終的にはクラスタの数は1個となるが、ここでは、画像処理装置が領域統合の過程においてクラスタの数が2個となった時点において領域統合を中止するようにする。

【0018】 この段階で領域統合を中止することで、与えられた画像の各画素を2つのクラスタに分類したことになる。このようにして得られた2つのクラスタについて見てみると、同一のクラスタを構成する画素同士は互いに色度が近似しているが、異なるクラスタ間の画素同士は色度が大きく異なっている。すなわち、画像データは特徴の異なる2つのグループに分割されたことになる。以上の処理により、画像を背景部分と人物部分の2つの領域に分割することが可能となる。

【0019】 図15は画像データおよび領域統合処理の概念図である。図15中の符号21は、入力された画像データのうちの一部分（9画素分）を示している。また、図16は $L^* a^* b^*$ 色空間上における画像データを表す図である。先行技術にあつては、例えばある画素（クラスタ）210と、他のいずれの画素を統合させるかを以下のように判断している。

【0020】 先ず、画像処理装置は、上記ある画素（以下、着目画素と呼ぶ）210に隣接する他の画素（四近傍（画素は方形として考えて、着目画素の四辺に接する4つの画素）または八近傍（着目画素の四辺に接する4つの画素と、着目画素における対角位置に接する4つの画素の計8画素））を探し出す。すなわち、着目画素210に対して隣接する画素のみが、当該着目画素210と統合可能な対象画素である。

【0021】 そして、画像処理装置は、着目画素210に隣接する他の画素のうち、対比強度尺度値が最小となる（類似度が最大となる）画素を探し出す。対比強度尺度値は、後述するように $L^* a^* b^*$ 色空間（図16）上において領域統合をした場合に、クラスタの分散値の

変化を表わしたものである。

【0022】 $L^* a^* b^*$ 色空間は均等色空間であり、この空間上における距離は視覚上の色度の差に略比例したものとされている。すなわち、 $L^* a^* b^*$ 色空間における対比強度尺度値が小さいということは、クラスタ対の色度等が近似している（類似度が大きい）ことを意味し、対比強度尺度値が大きいということは領域対の色度等が相違（類似度が小さい）ということの意味している。

【0023】例えば、 $L^* a^* b^*$ 色空間において着目画素 210 に対してその周囲の各画素について対比強度尺度値の大小を調べる。その結果、着目画素 210 に対して、画素 211 の持つ対比強度尺度値が最小であったとすると、画像処理装置は着目画素 210 とこの対比強度尺度値最小の画素 211 との領域統合を行う。そして、これが終わると再びこの統合された着目画素 210 に対するこのような処理を繰り返す。このような処理を繰り返す毎に、隣接する画素（クラスタ）対のうち、その時点での比較対象中における対比強度尺度値最小のものが着目画素に統合されてゆくことで、色度等の近似したものが順次統合されて、画素（クラスタ）対は一つずつ減少し、最終的には元の 9 個の画素は 1 つのクラスタ（領域）にまとめられることになる。

【0024】上述した階層的クラスタリング処理を効率良く行うために、先行技術ではいわゆるグラフ構造によるデータ処理を行っている。すなわち、画像データを構成する画素（クラスタ）を頂点とみなし、 $L^* a^* b^*$ 色空間における画素（クラスタ）間の対比強度尺度値を辺とみなし、各領域および各領域間の類似度等をグラフとして扱うものである。これらの頂点および辺の全体をグラフとしてとらえると、階層的クラスタリングの処理はいわゆる木（tree）を構成することになる。以下に、本画像処理装置における各種データ構造を図 18～図 21 を参照しながら説明する。

【0025】図 18 は、頂点のデータ 50 および辺のデータ 51 を表わしている。頂点のデータ 50 はクラスタの情報を表すものであり、以下のデータにより構成されている。同図において、ラベル 501 はグラフ中の一意の記号であれば良いが、本先行技術では頂点であるクラスタの実画像上の座標（ x, y ）を表わしている。例えば、32 ビットのデータのうちの上位 16 ビットを y 座標値、下位 16 ビットを x 座標値とすると、ラベル 501 は $y \times 65536 + x$ のように表わされる。また、画像の幅を w とし、 $x + y \times w$ のようにラベル 501 を表わすことも可能である。

【0026】 L^* 平均値 502、 a^* 平均値 503、 b^* 平均値 504 は、クラスタ内の各画素についての $L^* a^* b^*$ 色空間における各座標の平均値を表わしたものである。サンプル数 505 は、クラスタを構成する全画素数を表わしている。リストへのポインタ 506 は、コ

ンピュータにおけるメモリ上に置かれたリストデータ 60 のアドレスを表わすものである。また、リストデータ 60 は、辺のデータのポインタよりなるものであり、これについては後述する（図 19）。二分木へのポインタ 507 は、統合前の頂点を表わす二分木のノードを表わすものである。すなわち、二分木のデータはクラスタの統合履歴をツリー構造のデータとして表わしたものである。

【0027】辺のデータ 51 は以下のように構成されている。対比強度尺度値 511 は辺の両端にある頂点（クラスタ）間の $L^* a^* b^*$ 色空間上における分散変化を表わしている。すなわち、対比強度尺度値 511 は、対比される 2 つのクラスタを統合したと仮定して、統合前後における $L^* a^* b^*$ 色空間上のクラスタの分散値の変化を表わしたものである。

【0028】従って、“対比強度尺度値が小さい”と云うことは、2 つのクラスタを統合した場合における分散値の変化が少ないと云うことであり、これはすなわち、2 つのクラスタの色度が近似している（類似度が大きい）と云うことを意味している。

【0029】逆に、“対比強度尺度値が大きい”と云うことは、2 つのクラスタを統合した場合における分散値の変化が大きいと云うことであり、これは結局のところ、2 つのクラスタの色度が相違している（類似度が小さい）と云うことを意味している。なお、図 16 において、対比強度尺度値 d_{pq} の算出式に従い、対比強度尺度値を算出する際に、 $L^* a^* b^*$ の各成分に重み付けをしても良い。

【0030】辺のデータ 51 中の頂点のデータへのポインタ 512、513 は、辺の両端の頂点のデータのアドレスを示している。従って、辺のデータ 51 を参照することにより、クラスタの連結関係および各クラスタ間の類似度を容易に判断することが可能となる。

【0031】図 19 は、リストデータ 60、頂点のデータ 50、辺のデータ 51 の関係を表わしている。この図の頂点のデータ 50、辺のデータ 51 は上述した通りのものであり、また、リストデータ 60 は、辺のデータ 51 のアドレスを示すポインタ 601 より構成されている。よって、リストデータ 60 を参照することにより、辺および頂点の連結関係を把握することが可能となる。また、辺のデータ 51 中のポインタ 512、513 を参照することにより、辺の両端にある 2 つの頂点のデータ 50 を探し出すことができ、頂点のデータ 50 中のリストへのポインタ 506 を参照することによりリストデータ 60 を探し出すことができる。

【0032】図 20 は、頂点のデータ 50、ツリー構造のノードデータ 70 を表わしている。ノードデータ 70 には、2 つの頂点（クラスタ）を統合する際の情報が含まれており、二分木のデータの一部を構成するものである。ラベル 701 は統合後の頂点のラベルで、本先行技

術では、統合前の2つの頂点のラベルのいずれかとしてある。

【0033】 L^* 平均値702、 a^* 平均値703、 b^* 平均値704は、2つのクラスタを統合した後の各色成分の平均値である。すなわち、各クラスタの総画素数に従い各色成分の値を加重平均したものが、 L^* 平均値702、 a^* 平均値703、 b^* 平均値704である。対比強度尺度値705は、上述したように2つのクラスタを統合した場合における L^* a^* b^* 色空間上の分散値の変化を示している。

【0034】サブ・ツリーへのポインタ706、707は、統合されたノードデータ70を指し示すものである。このように階層的構造をなすノードデータ70を辿っていくことにより、領域統合の過程を把握することができる。

【0035】図21は、B-木 (tree) を表わす図である。このB-木80は、対比強度尺度値をキー (検索項目) としてリスト81~86、...の中から対比強度尺度の最小となる辺のデータを検索するために用いられる。すなわち、このB-木80を参照することにより、ある頂点 (クラスタ) と、この頂点に隣接した他の頂点とを結ぶ辺の中から色度が最も類似したものを、極めて少ないメモリアクセスで検索することが可能となる。

【0036】この図において、リスト81~86、...には辺のデータ51のアドレスを指し示すポインタが書き込まれており、同一リストのポインタで示される辺のデータ51は同一の対比強度尺度値をもつものである。これらのポインタは辺のデータ51中の対比強度尺度値511の昇順に対応して並んでいる。従って、最上位のリスト81に登録されたポインタを調べ、このポインタの示す辺のデータ51を探し出すことにより、対比強度尺度値が最小となる辺のデータ51を探索できる。なお、対比強度尺度値が同一となる辺が複数存在する場合には、同一のリストに複数のポインタが登録される。

【0037】B-木80における根のノードは4つのキー811~814からなり、さらにそれぞれのキーの前後に5つのポインタ821~825があり、各ポインタは一つのノードを指し、各キーは一つのリストをポイントする。これらは、全体として階層構造をなしている。根の5つのポインタ821~825はそれぞれ、キー811より小さいキー、キー811より大きくキー812より小さいキー、キー812より大きくキー813より小さいキー、...、キー814より大きいキーのようにデータが分割して割り当てられ、これらの各ポインタに接続された各ノードには、さらにデータが分割して割り当てられている。

【0038】従って、各層毎に最上位のノード801、802、803を辿っていくことにより、最小の対比強度尺度値を示す辺のデータへのポインタのリスト81を

探索することができる。なお、B-木80は、リスト81、...の追加、削除が容易であると云う利点を備えている。

【0039】また、同一リスト内に複数の対比強度尺度値が登録されている場合、このうちの 하나가削除されたとしても、リストが空になるまでそのキーは削除されることはない。すなわち、B-木80は常に平衡状態を保つため、高速の検索が可能となる。

【0040】[先行技術における階層的クラスタリング処理の具体例] 続いて、本先行技術における階層的クラスタリング処理、すなわち各クラスタを1つのクラスタに統合するまでの処理を説明する。図17は本先行技術における領域統合の処理を表わすフローチャートである。本先行技術を適用したコンピュータによる画像処理装置にはまず、画像データを入力して、これを画像メモリに記憶させる。ここで、図15に示される画像データ21が入力されたとする。実際の画像データはこれより遥かに画素数が多いが、ここでは話を簡潔にするために便宜的に3×3の画素の画像データ21を例に説明を進める。

【0041】ステップS41において、プロセッサは画像データ21に基づき、頂点のデータ50、辺のデータ51、ノードデータ70を算出することにより、いわゆる格子グラフを生成する (ステップS41)。このステップS41の処理を詳細に表わしたものが、ステップS411、S412のサブルーチンである。

【0042】次にまずステップS411において、画像データ21の各画素についての頂点のデータを算出する。頂点のデータ50は上述したように、ラベル501、 L^* a^* b^* 色空間における各色成分の平均値502~504、サンプル数505、リストへのポインタ506、二分木へのポインタ507により構成される。なお、この時点では領域統合は行われていないため、ラベル501は各画素の画像データ上の座標値となり、サンプル数505は“1”となる。また、各色成分の平均値502~504は画素毎の L^* a^* b^* 色空間上の座標値となる。

【0043】さらに、プロセッサは各画素について、二分木のノードデータ70を登録する。このとき、領域統合は行われていないため所謂サブツリーは存在せず、よってサブツリーへのポインタ706、707は空欄 (ゼロ) となる。また、クラスタの分散の変化を表わす対比強度尺度値705も空欄 (ゼロ) となる。

【0044】つぎにステップS412の処理に移る。このステップS412においては、辺のデータ51を以下の手順で算出する。まず、プロセッサは、画像データ21中の画素210に着目し、この着目画素210と、この画素210に隣接する (例えば4近傍) 画素211~214との間の L^* a^* b^* 色空間上における対比強度尺度値を求める。

【0045】すなわち、プロセッサはステップS411で算出された頂点データ50中の $L^* a^* b^*$ の各色成分502~504に基づき、着目画素210と画素211~214との各画素間における各色成分502~504の分散の変化を調べる。そして、これらの分散の変化を、対比強度尺度値511として着目画素210の辺のデータ51に登録する。また、辺のデータ51のポインタ512, 513には着目画素210の辺の両端にある頂点のデータ50のアドレスを書き込む。このようにして、着目画素210を中心としたこの着目画素210における4つの辺のデータ51が生成される。

【0046】このようにして生成された辺のデータ51は、図21のB-木80にその検索用のキー（検索項目）の情報と辺のデータへのポインタと共に登録する。すなわち、プロセッサは、対比強度尺度値の小さい順に辺のデータ51のポインタを登録してゆくことで、リスト81、…が完成する。以上のステップS411, S412の処理により、頂点および辺のデータよりなる初期格子グラフが完成する（ステップS41）。

【0047】これが終わると、総クラスタ数が“1”になったか否かを判断する（S42）。このとき、領域統合は行われていないため、画素数がそのまま総クラスタ数となる。従って、総クラスタ数が“1”でないから、ステップS42の判断結果はNOとなり、次の対比強度最小のクラスタ対（辺）の検出処理を行う（ステップS43）。つまり、ステップS43での処理は、B-木80を参照し、リスト81、…の中から対比強度尺度値が最小となる辺のデータ51を検索し、そして、検索が終了したならば、リスト81を先出し、または、後出し、もしくは何らかの規定に従ってリスト全体から削除すると云った内容である。

【0048】対比強度尺度値の最小となる辺のデータ51が検索されると、つぎにこの辺のデータ51の両端にある頂点を統合する処理を行う（ステップS44）。例えば、着目画素210と画素211との対比強度尺度値が最小であったとすると、着目画素210と画素211との統合を行う。これに伴い、プロセッサは頂点のデータ50、辺のデータ51、二分木のデータ70等の更新も実施する。

【0049】頂点のデータ50の更新処理は、ステップS43において統合された画素210, 211のそれぞれに対応する2つの頂点のデータ50を読み出し、いずれか一方の頂点のデータ50を他方の頂点のデータ50に統合することで行う。

【0050】例えば、着目画素210に係る頂点のデータ50を画素211に係る頂点のデータに統合したとすると、2つの画素210, 211が一つのクラスタに統合されるとその結果、統合後のクラスタの $L^* a^* b^*$ 色空間上における各色成分の平均値502~504は変動する。従って、統合されたクラスタにおける各色成分の

平均値502~504も、着目画素210に係る頂点のデータ50に登録する。

【0051】ここで頂点のデータ50中の二分木へのポインタ507の更新は、先ず、新たなノードデータ70を生成し、着目画素210の頂点のデータ50中のポインタ507を、新たなノードデータ70のポインタ706に、そして、画素211の頂点のデータ中のポインタ507を、新たなノードデータ70のポインタ707にそれぞれ登録することで行う。また、着目画素210のラベル501を新たなノードデータ70のラベル701に、また、画素210, 211の色成分502~504の平均値を新たなノードデータ70の色成分702~704に、画素210, 211を結ぶ辺のデータ51中の対比強度尺度値511を新たなノードデータ70の対比強度尺度値705にそれぞれ登録する。

【0052】上記辺のデータ51の更新処理は、画素210, 211の双方のリスト60から頂点210, 211に関する辺のデータを削除すると云った処理となる。これは2つの頂点を統合した場合、各頂点が保有していた情報が重複してしまうことがあることから、リストデータ60、B-木80中の重複したデータを削除する必要があるためであり、従って、画素210, 211を統合する第1回目の領域統合の段階においては、画素210, 211が重複して連結する辺が存在しないため、かかる処理は必要がない。

【0053】このようにして新たに統合された領域に対する非類似度の更新を、これに隣接するすべての領域について行う。

【0054】以上により、画素210, 211が一つの頂点（クラスタ）216に統合される。この後、ステップS42の処理に戻り、S42~S44の処理を繰り返し実行する。このようにして、画素データ21の領域統合の段階で、例えば図16に示されるようにそれぞれ複数の画素を有するクラスタP, Q, Rが生成されたとする。

【0055】この段階ではステップS42の判断ルーチンにおいて、まだ総クラスタ数が“1”でないから、これらのクラスタP, Q, Rの領域統合を行う処理が続く。すなわち、ステップS42の判断の結果、“NO”であるから、クラスタP, Q, Rの領域統合の処理を行うべく、ステップS43の処理に移り、ここでプロセッサはB-木80を参照し、リスト81、…の中から対比強度尺度値が最小となる辺のデータ51を検索する。

【0056】検索が終了すると、次にリストからその検索された他の登録を削除し、リストが空であれば、そのキーをB-木から削除する。対比強度尺度値の最小となる辺のデータ51が検索されると、この辺のデータ51の両端にある頂点を統合する（ステップS44）。例えば、クラスタPとクラスタQとの対比強度尺度値が最小であったとすると、クラスタPとクラスタQとの統合が

行われる。これに伴い、頂点のデータ50、辺のデータ51、二分木のデータ70等の更新も行う（ステップS44）。

【0057】つまり、頂点のデータ50の更新処理は、ステップS43において統合されたクラスタP、Qのそれぞれに対応する2つの頂点のデータ50を読み出し、いずれか一方の頂点のデータ50を他方の頂点のデータ50に統合すると云ったことで行う。そしてその結果、例えば、クラスタPに係る頂点のデータ50をクラスタQに係る頂点のデータに統合したとすると、2つのクラスタP、Qが一つのクラスタに統合された結果、統合後のクラスタの $L^* a^* b^*$ 色空間上における各色成分の平均値502～504は変動するから、統合されたクラスタにおける各色成分の平均値502～504を、クラスタQに係る頂点のデータ50に登録する。

【0058】また、頂点のデータ50中の二分木へのポインタ507については、先ず、新たなノードデータ70を生成し、着目画素210に係る頂点のデータ50中のポインタ507を新たなノードデータ70のポインタ706に、画素211に係る頂点のデータ中のポインタ507を新たなノードデータ70のポインタ707にそれぞれ登録し、また、着目画素210に係るラベル501を新たなノードデータ70のラベル701に、画素210、211に係る色成分502～504の平均値を新たなノードデータ70の色成分702～704に、画素210、211を結ぶ辺のデータ51中の対比強度尺度値511を新たなノードデータ70の対比強度尺度値705にそれぞれ登録することで更新される。

【0059】辺のデータ51については、先ず、画素210、211の双方のリスト60から頂点210、211に係る辺のデータを削除すると云った更新処理をする。これは2つの頂点を統合した場合、各頂点が保有していた情報が重複してしまうと云った事態が生じることがあるためであり、そこで、リストデータ60、B-木80中の重複したデータを削除する。しかし、画素210、211を統合する第1回目の領域統合の段階においては、画素210、211が重複して連結するような辺が存在しないため、このような処理は必要がない。従って、新たに統合された領域に対する非類似度の更新を、これに隣接する全ての領域について行う。

【0060】以上により、画素210、211が一つの頂点（クラスタ）216に統合される。この後、プロセッサの処理はステップS42に戻り、ステップS42～S44の処理を繰り返し実行する。

【0061】このようにして、画素データ21の領域統合の段階で、例えば図16に示されるようにそれぞれ複数の画素を有するクラスタP、Q、Rが生成されたとする。これらのクラスタP、Q、Rについて次に再び領域統合の処理を次のようにして行う。

【0062】まず、ステップS43においてはB-木8

0を参照し、リスト81、…の中から対比強度尺度値が最小となる辺のデータ51を検索する。検索が終了すると、次にリストからその検索された他の登録を削除し、リストが空であれば、そのキーをB-木から削除する。対比強度尺度値の最小となる辺のデータ51が検索されると、この辺のデータ51の両端にある頂点を統合する。例えば、クラスタPとクラスタQとの対比強度尺度値が最小であったとすると、クラスタPとクラスタQとの統合が行われる。これに伴い、プロセッサは頂点のデータ50、辺のデータ51、二分木のデータ70等を更新する（ステップS44）。

【0063】頂点のデータ50中の二分木へのポインタ507の更新は、先ず、新たなノードデータ70を生成し、クラスタPに係る頂点のデータ50中のポインタ507を新たなノードデータ70のポインタ706に、クラスタQに係る頂点のデータ中のポインタ507を新たなノードデータ70のポインタ707にそれぞれ登録すると云った処理をなすことで行われる。また、クラスタPに係るラベル501を新たなノードデータ70のラベル701に、クラスタP、Qに係る色成分502～504の平均値を新たなノードデータ70の色成分702～704に、クラスタP、Qを結ぶ辺のデータ51中の対比強度尺度値511を新たなノードデータ70の対比強度尺度値705にそれぞれ登録する。

【0064】辺のデータ51の更新処理は図19に示すように、先ず、頂点P、Qを結ぶ辺PQを示すポインタ601をリストデータ60から削除し、また、頂点P、Qがともに連結する頂点Rが存在する場合、CPUは頂点P、Qに関するリストデータ60から、辺QRまたは辺RPのポインタ601を削除し、B-木80から辺QRまたは辺RPに関するデータを削除し、さらに、頂点Pに連結する辺のポインタ601を頂点Qのそれにマージすると云った処理である。このような処理により、重複したデータの削除等が行われ、辺のデータ51に関する更新処理が終了する。

【0065】さらに、原点P、Qと連結していた他の頂点Tとの辺のデータ51中の対比強度尺度値511は、統合後のPおよびTの $L^* a^* b^*$ の平均値と構成画素数に基づき更新処理される（図16中段の式）。この処理は、上述した頂点データ50の更新処理におけるものと同様である。統合後の新たな対比強度尺度値511の更新に際し、更新されるべき辺のデータは一時B-木から削除され、対比強度尺度値511が更新されると、プロセッサは新たな対比強度尺度値511をキーとしてデータ51へのポインタをB-木80に登録する。

【0066】以上により、ステップS44の処理が終了する。この後、処理はステップS42に戻り、画像データ21の総クラスタ数が“1”になるまでステップS42～S44の処理を繰り返し実行する。そして、総クラスタ数が“1”になった時点で、領域統合処理は終了す

る。この結果、画像データ 21 は 1 つの頂点（クラス）250 に統合される。

【0067】なお、上述した領域統合処理は、一つのクラスに他のクラスを順次統合させるものであるが、領域統合を画像データの局所において並列に行うことも可能である。このような並列処理は特に初期の統合過程において有効である。

【0068】図 17 のフローチャートで表わされた手順による領域統合処理が終了すると、次に統合履歴を表わす二分木のノードデータ 70 を、外部記憶装置に書き込んで保存する。なお、二分木を構成するノードデータ 70 の数は画像データの総画素数の約 2 倍となることから、二分木のデータ数は膨大な数に達する。従って、二分木のデータの全ての二分木探索を行っていたのでは、この探索に費やす負担が重過ぎる。

【0069】そこで、本先行技術においては、二分木を構成するノードデータ 70 毎に、これを二分木探索の順に外部記憶装置に書き込み、読み出し時にはこれと逆の順序でノードデータ 70 毎に外部記憶装置から読み込んで探索に利用する。このようにすることにより、二分木再配置のための負担を大幅に低減することが可能になる。

【0070】外部記憶装置に対するルートノードからの書き込みおよび読み込み時の処理を図 20 に基づき説明する。

【0071】書き込み時においては、ルートノードまたは所定のノードを始点として以下の処理を再帰的に実行する。まず、始点となるノードデータ 70 中のラベル 701、 $L^* a^* b^*$ の各平均値 702~704、対比強度尺度値 705 を外部記憶装置に書き込む。次に、このノードデータ 70 サブ・ツリーへのポインタ 706 を参照し、サブ・ツリーへのポインタ 706 が空欄でなければ、サブ・ツリー 706 が示す次のノードデータ 70 を外部記憶装置に書き込む。ポインタ 706 が空欄である場合には、このポインタ 706 が書き込まれたノードデータ 70 が「葉」である旨を外部記憶装置に書き込んでから一つ上の親ノードに戻り、ポインタ 707 についても同様の処理を行う。以上の処理を繰り返すことにより、ノードデータ 70 を二分木探索の順に外部記憶装置に書き込むことができる。

【0072】なお、上記書き込み処理において、二分木の「葉」となるノードデータ 70 の判別をサブ・ツリーへのポインタ 706、707 が空欄か否かにより行っているが、対比強度尺度値 705 が空欄か否かにより行っても良い。また、ノードデータ 70 に対応する頂点データ 50 中のサンプル数 505 が「1」になったことをもって当該ノードデータ 70 が「葉」であることを判断しても良い。

【0073】続いて、ノードデータ 70 の読み込み処理を説明する。まず、画像処理装置では作業領域となる自

己のワークメモリ上にノードデータ 70 の領域を確保し、ここに外部記憶装置から読み出された始点となるノードデータ 70 中の各データを転送する。次にワークメモリから読み出されるノードデータはこのノードデータ 70 のポインタ 706 がポイントするものとして新たなノードデータ 70 を確保し、そこにワークメモリからのデータを読み込む。このとき、読み込まれたデータが「葉」でなければ、ノードデータ 70 のポインタ 706 について同様の処理を繰り返す。「葉」である場合には、このノードデータ 70 のポインタ 706、707 を空欄として一つ上のノードに戻り、そのノードのポインタ 707 にポイントされるものとして新たなノードデータ 70 の確保と、ワークメモリからのデータの読み込みを行い、このノードデータ 70 のポインタ 706 について処理を継続する。

【0074】続いて、この二分木を利用した領域抽出の処理について説明する。上述した領域統合処理により、図 15 の (a) に示されるように画像データは最終的に 1 つのクラス（頂点）250 に統合された。この領域統合の過程を逆に辿ることにより、図 15 の (b) に示されるように、画像データ 21 を分割した画像データ 22 を得ることが可能である。すなわち、背景、人物等を表す画像データを、背景の画像領域、人物の画像領域等に分割し、所望の画像領域を抽出することができることになる。

【0075】つまり、二分木（履歴データ）をルートノードから「葉（リーフ）」に向かって辿っていくことにより、1 のクラスを順次 2 つのクラスに分割することができるわけであるが、二分木をどの深さまで辿るかを以下の手順で予め決定しておく。

【0076】まず、分割を終了させる条件として対比強度尺度値の下限（類似度の範囲）を、予め設定しておく。すなわち、二分木に従い一つのクラスを分割していく過程において、各クラスの色度がある程度近似したならば分割処理を中止させるためである。この対比強度尺度値の下限（類似度の範囲）を定めてこの下限で分割を終了させることにより、画像を特徴の全く相違する領域毎に分割することができるようになる。

【0077】対比強度尺度値の下限値が定められると、これを基準として対比強度尺度値の大小判断を行う。すなわち、ルートのノードデータ 70 中の対比強度尺度値 705 が指示された下限値より小さいか否かを判断する。このとき、対比強度尺度値 705 が下限値より小さい（類似度が所定値以上）場合には処理を終了する。

【0078】一方、対比強度尺度値 705 が下限値よりも大きい（類似度が所定値以下）場合には、ノードデータ 70 中のポインタ 706、707 を参照し、そして、ポインタ 706、707 が空欄である場合（すなわち、サブ・ツリーが存在しない場合）には、処理を終了する。ポインタ 706、707 が空欄でない場合には、ポ

インタ 706, 707 で示されたサブ・ツリーのノードデータ 70 を参照する。そして、このサブ・ツリーの対比強度尺度値 705 が下限値よりも小さい場合には処理を終了する。

【0079】これらの処理を繰り返すことにより、二分木が限定される（二分木の枝が途中で切断される）。そして、このようにして限定された二分木の「葉」によって示される各クラスタは、ある程度特徴が相違したものとなっている。

【0080】図 15 の (b) に示されるように、ノード 250 をルートとする二分木は画像データ 21 の各画素に向かって延びているが、この二分木を途中で切断することにより、4 つのクラスタを「葉」とする二分木が得られる。

【0081】このようにして限定された二分木の各葉を、それ以下のサブ・ツリーのルートノードとして、分割された領域を示す領域指定データを生成する。なお、この領域指定データは画像データ 21 に対応した座標値を持つ二次元データである。この領域指定データ生成処理は次のようにして行う。まず、ルートノードのポインタ 706, 707 を参照する。そして、ポインタ 706, 707 が空欄である場合には、ラベル 701 から実画像における x, y 座標を算出し、この x, y 座標に対応した領域指定データの座標にラベル 701 を書き込む。一方、ポインタ 706, 707 が空欄でない場合には、これらのポインタ 706, 707 で示されたサブ・ツリーのノードデータ 70 を参照し、上記と同様の処理を繰り返していく。

【0082】このような処理を行うことにより、上述の分割された各領域にラベル 701 が付される。そして、各領域に付されたラベル 701 を基に、画像データ 21 から所望の領域を抽出することが可能となる。従って、上述の人物と背景とを表す画像データから、人物を表す領域のみを抽出すると云ったことができることになる。

【0083】このような手法の先行技術で画像処理を行うことにより、領域分割された画像の一例を図 22 および図 23 に示す。図 22 は原画像を表わしている。図 23 はこの原画像を領域分割したデータを表わしている。図 22 の原画像について階層的クラスタリングを行うことにより、図 23 の (a)、(b)、(c) のように、順にクラスタが統合されていく。

【0084】図 24 は、図 22 の原画像を領域統合する過程を表わしたツリーである。このツリーの横軸は領域統合の非類似度を対数で表わしたものである。また、図 24 に付された矢印 (A)、(B)、(C) の各段階の画像が、図 23 の (a)、(b)、(c) にそれぞれ対応している。

【0085】図 26、図 27、図 28 は、領域統合の各段階における頂点および辺を示すグラフであり、それぞれ図 23 の (a)、(b)、(c) に対応していて、そ

れぞれ領域統合の各段階における頂点および辺を示している。図 26 は図 24 の矢印 (A) の段階のグラフであり、図 27 は図 24 の矢印 (B) の段階のグラフである。さらに、図 28 は図 24 の矢印 (C) の段階のグラフを表わしている。

【0086】図 25 は、原画像 121 (図 22 に示されたものと同じ) を領域統合する各段階での画像 121a ~ 121f を示している。この図の画像 121c, 121d, 121e は、図 23 の (a)、(b)、(c) に対応している。

【0087】以上、本発明者の提案した先行技術としての階層的な領域抽出方法の詳細を説明した。そして、この先行技術では、隣接するクラスタ対のうち、色空間上の対比強度尺度値の最小となる（類似度の最大となる）もの同士を統合することにより、階層的クラスタリングを行っている。従って、領域統合の判断を隣接するクラスタについてのみ行えば良いため、処理時間を大幅に短縮できるとともに処理に要するメモリ容量を軽減できる。

【0088】また、 $L^* a^* b^*$ 色空間上の分散変化の最小となるクラスタ同士を統合するため、大局的な対比関係を捉えると云う視覚特性に合致した領域分割が可能となる。例えば、写真、映像、印刷等の分野においては、階調を「ライト、中間、シャドウ」のように主観的に分類することがある。この先行技術によれば、視覚特性に合致した $L^* a^* b^*$ 色空間を用いて領域分割を行っているため「ライト、中間、シャドウ」のような主観的な画像の分割が可能となる。

【0089】

【発明が解決しようとする課題】従来知られている画像処理方法のいずれもが、画像を「ライト、中間、シャドウ」の各領域に分割する処理に適用するには不向きであり、この問題を解決するために本発明者は、特願平 5-340077 号に示す如き階層的な領域抽出方法を提案した。そして、この先行技術においては画像処理に階層的クラスタ分析法を使用している。

【0090】一般的に、階層的クラスタ分析法の画像への適用には、その処理にあたって計算機の記憶資源を大量に必要とし、通常の電子計算機ではその処理実行に関して、非現実的な記憶容量が必要とされる。

【0091】すなわち、階層的領域抽出の実行を、全画面に対して一回で終了させようとする、グラフ構造により、隣接する領域関係の管理だけをしても、相当量の記憶資源が必要となる。

【0092】四近傍隣接の管理だけをするとともに、隣接関係の組み合わせは、縦横が $h \times w$ 画素の画像においては、

$$2hw - (w + h)$$

となる。これは、全ての画素の組み合わせ（非類似度行列を用いる場合）を取り扱えば、

$$hwC_2 = hw(hw-1)/2$$

となり、明らかに前者は、発生し管理しなければならない画素対の最大が、総画素数に比例し、後者はその最大数が総画素数の二乗に比例する。それでも、グラフ構造を用いて実現しようとする場合、一つのクラスタあたり200から300Byteの記憶容量を必要とし、512*512程度の画像においても100MByte近くの記憶容量が必要となる。

【0093】膨大なデータを電子計算機で処理する場合、処理速度を重視するならば、データを電子計算機の主記憶（メインメモリ）におくことが望ましい。そして、主記憶として、上述のような記憶容量のメモリを確保することは、実現性と云う視点でとらえれば、全く不可能な数字ではないが、その一方で現時点における記憶資源のコストが高価であると言う現実があることから、これ程の記憶容量を主記憶として用意するには経済的に制約が大きい。また、主記憶の代わりに仮想記憶を用いることも可能であるが、外部記憶とのデータの授受は膨大なオーバーヘッドとなり、処理時間が急激に増大することとなるから、実用性の面で問題が大きい。

【0094】また、グラフ構造により近傍隣接する画素の対のみを管理するようにする場合、全ての画素の対の非類似度の情報はなため、それで、再帰的に非類似度を算出する方法（組み合わせ的方法）の適用範囲が限定される。

【0095】この問題を解決するため、上述した先行技術では、計算機記憶資源の必要量が処理を行う画像の総画素数に比例して使用されるようにするために、グラフ構造を利用するようにした。これにより、電子計算機での処理に適合するようになったが、これであっても記憶容量の小さな電子計算機で処理をしようとするのを考えると、記憶容量あるいは実行速度の点から支障がある。

【0096】そこで、本発明の目的とするところは、階層的クラスタ分析により、画像の画面上の色の対局的な対比関係によって相互に分化し成立する階層的な領域構造を抽出するにあたり、比較的記憶資源の小さな電子計算機においても処理可能にする画像処理方法を提供することにある。

【0097】

【課題を解決するための手段】上記目的を達成するために本発明は次のようにする。すなわち、第1には処理対象の画像を所要の単位に分け、この分けた画像を単位に、その構成画素について領域統合する第1の領域統合処理工程と、この第1の領域統合処理工程により得られた上記各単位の画像を取り纏めてこれらを初期クラスタとする1画面分の画像とし、この画像のクラスタについて領域統合を行うとともに、統合履歴を保存する第2の領域統合処理工程とよりなることを特徴とする。

【0098】また第2には、処理対象の画像を所要の単

位に分け、この分けた画像を単位に、その構成画素について領域統合する第1の領域統合処理工程と、この第1の領域統合処理工程により得られた上記各単位の画像を取り纏めてこれらを初期クラスタとする1画面分の画像とし、この画像のクラスタについて領域統合を行うとともに、領域統合の履歴情報を二分木により管理すべく情報保存する第2の領域統合処理工程とよりなることを特徴とする。

【0099】さらには、領域統合の履歴は、領域統合の順序および領域統合された領域間の類似度を表わす情報、統合されるクラスタの座標情報を含むと共に、二分木で管理することを特徴とする。

【0100】

【作用】本発明では、処理対象の画像を適宜な単位に分け、この分けた画像を単位に、その構成クラスタについて領域統合する第1段階での領域統合を行って構成クラスタを減少させ、この第1段階での領域統合が済んだ後、この統合処理済みのクラスタを取り纏めて元の1画面分の画像として扱い、第2段階での領域統合を行うとともに、統合履歴を保持することで目的の領域の抽出を可能にする。

【0101】第1段階での領域統合は画素数つまりクラスタ数を少なくするための前処理に相当し、この段階で得られたものを初期クラスタとして第2段階での領域統合を行うことで、各段階において少ないデータ数で処理ができるようになり、記憶資源の限られた計算機において十分処理を可能にしておき、また、統合の履歴をその時々各クラスタの色成分情報とともに保存しておくので、この履歴を辿ってゆくことで目的の領域の画像抽出を可能にする。

【0102】非類似度列を用いるにせよ、グラフ構造を用いるにせよ、階層的クラスタ分析法により、画像の画面上の対比により相互に分化成立する階層的領域の抽出は、計算機利用による処理上、膨大な記憶資源を必要とする。そこで、本発明では処理対象の画像を分割あるいはブロック化して領域統合処理を行うことで、領域統合処理毎の一度に取り扱う画素・クラスタ・領域の個数を限定することができようし、これによって記憶資源の小さな電子計算機においても処理が可能になるようにした。

【0103】

【実施例】以下、本発明の実施例について図面を参照して説明する。

【0104】非類似度列を用いるにせよ、グラフ構造を用いるにせよ、階層的クラスタ分析法により、画像の画面上の対比により相互に分化成立する階層的領域の抽出は、計算機利用による処理上、膨大な記憶資源を必要とする点を改善するために、本発明では処理対象の画像を分割あるいはブロック化して領域統合処理を行うことで、領域統合処理毎の一度に取り扱う画素・クラスタ・

領域の個数を限定することができようにし、これによって記憶資源の小さな電子計算機においても処理が可能になるようにしている。

【0105】例えば、画面を $32 \times 32 (= 1024)$ 画素の単位となるブロックに分割するとする。非類似度行列として発生する画素の対の組み合わせは、 523776 個となり、画素対1あたりのデータ量が仮に 50Byte （非類似度行列のほうデータ構造はグラフ構造に比して簡単であるので、処理単位あたりに要する記憶資源は小さい）とすれば、この行列を主記憶上に構成するのに必要とされる記憶容量は、約 25MByte となり、標準的な計算機の記憶資源の大きさとなる。また、非類似度行列を用いる場合は、処理単位、即ち画素数が少なくなれば、二乗に比例して画素対の組み合わせが減少する。この条件で、グラフ構造を用いた場合は、初期の画素対の個数は 1984 個、記憶資源の必要量は、約 0.55MByte となる。

【0106】各ブロックでの領域統合は、完了させるのではなく、非類似度またはクラス数等が適当な条件に達した所で停止保留し、これらの統合履歴を保存し、この処理を他の全てのブロックに対して実行する。全てのブロックに対してこの処理が終了した段階で、任意のブロックの任意の領域を他のブロックの他の領域との関係をしらべ、全画像については、再度、グラフ構造または非類似度行列を構成し、階層的領域統合の処理を再度実行して、領域統合を完成させる。

【0107】各ブロックでの領域統合の停止は、2回目の領域統合である第2の領域統合処理工程を実施するに際して、画像全体でグラフ構造または非類似度行列が組める程度の領域数に減少していること、また、各クラス対の非類似度の分布ができるだけ均一となっているようにすることがポイントとなる。

【0108】また、本発明の実施例では、画像を正方形のブロックの単位で分割した例で説明するが、画像の画面の局所局所に適合してブロックは任意の矩形としても構わなければ、不定形であっても構わない他、また、ブロックが一つで一度に階層的領域抽出が実現できる場合も本発明方法の変形例としての地位を有するものであることを予め断っておく。

【0109】以下、本発明の実施例の詳細を説明する。本実施例に係る画像処理方法を実行可能な画像処理装置のブロック図を図1に示す。この画像処理装置はコンピュータを利用するもので、操作部9、バス10、CPU（プロセッサ）11、入力I/F（インタフェース）12、プログラムメモリ13、ワークメモリ14、画像メモリ15、出力I/F（インタフェース）16、外部記憶装置17、GDC（グラフィック・ディスプレイ・コントローラ）18、ディスプレイ19等により構成されている。

【0110】操作部9は、キーボードあるいはジョイス

ティック、マウスなどと云ったシステムに操作入力を与えたり、指令を与えたりするためのマンマシンインタフェースであり、条件設定や、二分木を辿ると云った操作等にも使用される。

【0111】CPU11は、プログラムメモリ13に記憶されたプログラムに従い画像の領域分割等の処理を実行するものである。入力I/F12は、処理対象となる画像データを本画像処理装置に入力するためのインタフェースである。画像データは自然画像を表わし、例えば 512×512 画素のRGB（Rはレッド、Gはグリーン、Bはブルー）（またはCMYK（Cはシアン、Mはマゼンタ、Yはイエロー、Kはブラック）、あるいはCIE LAB）のカラーのデータより構成される。

【0112】プログラムメモリ13は、上述したCPU11の処理手順を表すプログラムデータを格納するためのものであり、CPU11はこのプログラムメモリ13の格納プログラムを実行して目的の処理を行う。

【0113】ワークメモリ14はCPU11が領域分割の処理に伴い算出するデータ（後述するツリーデータ、リストデータ等）を一時保持しておくためのものである。また、画像メモリ15は入力I/F12から入力された画像データを蓄えるためのものである。

【0114】出力I/F16は、領域分割された画像データ等を外部機器（プリンタ等）に出力するためのデジタルインタフェースであり、また、外部記憶装置17は、ハードディスクユニット、光磁気ディスクユニット等により構成され、画像データ、領域分割された画像データ、ツリーデータ、リストデータ、プログラムデータ等を保存する大容量記憶装置である。

【0115】GDC18は画像データおよび領域分割後の画像データ等に基づき、表示データを生成するものであり、ディスプレイ19はこの表示データに基づく画像を表示して操作者に提示するためのものである。

【0116】本実施例に係る画像処理方法の詳細を説明する。本発明の実施例では、画像を階層的クラスタリングするにあたり（つまり、例えば、自然画像における画面上の色の分布の大局的対比の階層的関係・構造を抽出するにあたり）、隣接するクラス対のうち、色空間上の対比強度尺度値の最小となる（類似度の最大となる）もの同士を次々に統合するが、ここでの階層的クラスタリングは処理すべき1枚の画像について、図2に示すようにまず第1段階として、処理し易い大きさに分割することにより複数の分割画像とし、処理すべき画素数を少なくして扱い易いようにする（S110）。そして、個々の分割画像についてそれぞれ階層的クラスタリングを行う（S120）。

【0117】そして、個々の分割画像についてそれぞれ階層的クラスタリングが成された段階で、次に第2段階の処理として、階層的クラスタリングが成されたこれら分割画像全体を取り纏めて1枚の画像として扱い、つま

り初期クラスタとして用い、これを階層的クラスタリングする (S130, S140)。その際、統合の履歴をその時々クラスタの色成分情報と共に保存する。そして、統合の履歴を辿り、所望の色成分の情報を呈するクラスタを知ってこれを抽出することにより、画像の所望の領域部分を分割することができるようにする。

【0118】一枚の画像を小分けしてクラスタリングしてから、クラスタリング済みの分割画像を一枚の全体画像に取り纏め、初期クラスタとし、再びこれをクラスタリングすると云った手法を採用することにより、一度に処理しなければならないデータ量を大幅に少なくして、有り余る程にはメモリ資源を持たないパソコンなどの装置を画像処理装置として使用した場合でも、画像中の目的像領域を分割することができるようにする。

【0119】本発明では、階層的領域抽出に当たって、グラフ構造を用いた階層的領域抽出法と、非類似度行列を用いての階層的領域抽出法等が利用できるが、グラフ構造を用いた階層的領域抽出法に関しては既に述べたので、ここでは非類似度行列を用いての階層的領域抽出を例に本発明を説明する。尚、処理対象画像の画像データは、自然画像、色空間は任意であるが、ここでは、CIE LAB表色の画像を用いるものとする。

【0120】本発明では入力された処理対象画像に対して、第1段階の領域統合処理においては、操作部9により操作者が設定したブロックサイズ設定情報に従うか、あるいは後述する条件に従って、CPU11は図2に示すように画像を処理し易いサイズに分割する (S110)。1枚の画像Fを一定サイズの矩形でブロックBに分割して分割画像化した例を図5に示す。この例では各ブロック (各分割画像) Bの矩形のサイズを32×32画素とする。従って、各ブロック (各分割画像) Bの総画素数はそれぞれ1024画素となる。

【0121】図6は各ブロック (各分割画像) Bの詳細を示したものである。また、図4は、本発明の処理フローの概略である。カラム、ラインの相対座標とブロック内での非類似度行列のインデックスを示してある。

【0122】また、1枚の画像を等ブロックサイズに分割してゆくと、そのサイズに満たない矩形が生じることがある。この場合、画像の縁辺部が問題にならないのであれば、ブロック群の位置を適当にセンタリングするなり、端物となったブロック (分割画像) を無視するようにすることも可能であり、また、1画面の縁片部に位置するブロック (分割画像) については上記ブロックサイズに満たない半端ブロックであるとして処理するようにしても構わない。また、この場合であれば、縦横各辺を、各ブロックの総画素数が1024を越えないように等分割し、全てのブロックのサイズを揃えることなどの方法が考えられる。

【0123】本発明の実施例に係る画像処理方法は、全体として、階層的領域統合を2段階で実現する手法とす

るが、その第1段階目の領域統合処理として、1画面の画像のブロック化を終えたならば、各ブロック (各分割画像) 単位での暫定的な領域統合を行う (図2のS120)。

【0124】ステップS20において実施される暫定的な領域統合の処理は、図3(a)に示すように、まず初めに、統合リストを初期化し (S121a)、全ブロックの統合処理が終了したか否かを判定して (S121b)、その結果、終了していなければブロックiについてそのクラスタの領域統合処理を実施する (S122, S123)。ブロックiについてそのクラスタの領域統合処理を終了したならば、iを更新し (つぎのブロック)、統合処理が未終了のブロックがあるか、否かをチェックする (S121b)。その結果、統合処理が未終了のブロックがあれば、更新したブロックiについてそのクラスタの領域統合処理を実施する (S22, S123)。そして、S121bのチェックをする (S121b)。その結果、統合処理が未終了のブロックがなくなれば、統合処理を終える。

【0125】ステップS123において実施したブロックiについてのクラスタの領域統合処理の詳細を図3(b)に示す。

【0126】領域統合処理はまず各画素毎にインデックスを振り、1画素1クラスタとして扱えるようにする (S123-1)。つぎに、初期非類似度行列の生成を行い、各画素間 (クラスタ間) で、非類似度 $d_{ij}(x, y)$ 、 $d(1^* a^* b^*)$ を求める (S123-2)。つぎに、ブロック間領域数は k_i になったか否かを調べ (S123-3)、なっていないならば $d_{ij}(x, y) < 1.2^{1/2}$ で $d(1^* a^* b^*)$ が最小の画素対 (クラスタ対) を検出し、統合処理を行う (S123-4)。

【0127】そして、非類似度行列の更新を行い (S123-5)、つぎに、ブロック間領域数は k_i になったか否かを調べ (S123-3)、なっていないならばS123-4以降の処理を繰り返す。ステップS123-3の判定の結果、ブロック間領域数は k_i になったならば領域統合履歴を全てリストに登録する (S123-6)。

【0128】以上が各ブロック (各分割画像) 単位での暫定的な領域統合処理である。ここで、当該暫定的な領域統合を行うことにより、1画面分の画像全体では1024個の領域に統合するものとする。1画面分の画像全体で1024個の領域に統合されたならば、CPU11は第2段階目の領域統合処理に入る。

【0129】第2段階目の領域統合処理は、この第1段階での領域統合処理により得られた1024個の領域を初期クラスタとして利用し、当該1024個の領域を取り纏めて元の1画面分の画像とし、これら1024個の領域を初期クラスタとして、階層的統合をする。

【0130】本発明では各段階での処理を、このような

1024個のデータについて行うようにすることにより、少ないメモリ資源の画像処理系で処理することを可能にしている。しかし、これは実施例としての一例であって、もちろん、扱うデータ数としての1024個に限定されるものではない。

【0131】ここで、第1段階での各ブロックの最終領域統合数（最終的にそのブロックで残る領域の数）の割り付けは、単純に、1画面の画素数をこの総ブロック数1024で割り、均一に分配するもので良いが、1画面の像の複雑さを考慮して、各ブロックはそれぞれの位置に応じた絵柄の複雑さを評価して、それに応じた最終領域統合数に暫定的に配分する方法等が考えられる。例えば、各ブロックの分散を求め、これらブロック毎の分散の総和で各ブロックの分散を割り、その値を、この場合では、1024に対するそのブロックの暫定的な領域統合の配分の比として利用する。そして、これにより得られた数を、各ブロックの暫定的な領域統合の数として使用するが、適当な整数に丸め直す方法なども考えられる。この時、ブロックに割り当てられる最終領域統合数は最低でも“1”でなければならないものとする。

【0132】ここでは、また、非類似度を揃えるのであれば、各ブロック中最小の非類似度を有するクラス対を検出し、これら対の最小の非類似度を有するものから統合してゆき、各ブロック内の領域数の総和が、この場合1024に達するまで行うようにすることも考えられる。しかし、これは基本的に計算効率が極めて悪くなる可能性が高い。そこでこれを回避するには、暫定的に目標となる非類似度を設定し、各ブロック毎にこの非類似度以上のクラス対が現われるまで階層的領域統合を実行し、最後に各ブロックの領域を総和して、1024を越えている場合は、再度、目標非類似度を高めに設定し、最後に画像における総クラス数が1024を越えなくなるまで実行する方法が考えられる。

【0133】ここでは、各ブロックの分散が画面上での画像の色の対比によって分化成立する階層的領域の複雑の尺度あるものとして、上記第1段階の領域統合を実現するものとする。

【0134】1画面分の画像データが入力I/F12を介して入力されると画像処理装置のCPU（プロセッサ）11はこれを画像メモリ15に取り込み、そして、この取り込んだ画像データについて、上述したような第1段階の領域統合処理を行う。この第1段階の領域統合処理を開始するに当たり前処理を行うが、この前処理は、上記画像データのブロック分割である（S110）。そして、ブロックに分割されるとつぎに第1段階の領域統合処理に入る（S120）。

【0135】【第1段階の領域統合処理】ここでの処理（S120での処理）は、基本的に順次各ブロックにおける領域処理を対象にしており、一つのブロックの領域統合処理が終わると次の別なブロックの領域統合処理に

移るといったループをなす。各ブロックにおける領域処理では、まず図6に示すように各画素にインデックスまたはラベル付を行う。インデックスについては非類似度行列の成分に対応するので“1”から始まる連続する整数を順次割り振り、また、ラベル付けについてはここでは相対的なラスタの座標値を用いるようにした。

【0136】ラベルはブロック内で一意である他、画像全体でも一意であるように、ここでは、画像全体に対する各画素の座標値より定めるものとする。例えば、図6に示すようにラベル190が32ビットのデータであるとすれば、その上位16ビットにラインの座標値、下位16ビットにカラムの座標値をセットすると云った手法をとる。

【0137】（非類似度行列生成）ブロック内の各画素に対するインデックスまたはラベル付を終えると、CPU11は次に非類似度行列を生成する。

【0138】非類似度行列は、図7に示すような2次元配列120とした。そして、配列の成分121の値は、上記ブロックにおけるクラス対の座標値（インデックス）の対に対応するように定める。各配列成分は画素対（クラス対）の情報を有する構造体130へのポインタである。この情報構造体130を図8（a）に示すように、クラス対Pの状態情報構造体へのポインタ情報部131、クラス対Qの状態情報構造体へのポインタ情報部132、色空間上でのP、Qの非類似度情報部133、画面上でのP、Qの非類似度情報部134にて構成する。

【0139】非類似度行列生成にあたり領域統合課程の初期条件として、第1段階の領域統合処理においては1画素1クラス対であるとして扱う。そして、この場合、構造体130のメンバは、対をなすクラス対それぞれのクラス対の状態を示す構造体140へのポインタ情報を保持させるための情報部131、132と、このクラス対における非類似度の情報を保持させるための情報部133、134で構成してある。

【0140】また、非類似度の情報は画面上の距離に関する非類似度を情報部134に保持させ、色空間分布に関する非類似度を情報部133に保持させる。上記画面上の距離に関する非類似度情報部134に格納される非類似度情報は、クラス対間の画面上の距離に関する非類似度を示すものであり、色空間分布に関する非類似度情報部133に格納される非類似度情報は、クラス対間の色の類似性に関する非類似度を示すものである。

【0141】画面上の距離に関する非類似度としては、ここでは、隣接関係の尺度を表わし、この場合、最小距離法として定められるものとする。また、色空間分布に関する非類似度は色の類似性の尺度となるが、ここでは、Ward（ワード）法により与えられるものとする。

【0142】また、各クラス対の状態の情報を保持する

構造体140は図8(b)に示すように、そのクラスタのラベルを格納したラベル情報部141と、サンプル数(画素数 n)を格納した情報部142、そしてこの例の場合にはさらに、各色成分 L^* 、 a^* 、 b^* の平均値情報を格納した情報部143、144、145と統合履歴へのポインタ情報を格納した情報部146からなり、これらの情報を保持する。

【0143】非類似度行列生成は、初期の非類似度行列の生成の場合、図3(b)に示す初期の非類似度行列の生成ステップS123-2において行われる。この初期の非類似度行列の生成ステップS123-2においては、クラスタの状態に関する構造体140及びクラスタ対に関する構造体130に、各画素および画素対の情報を書き込み、そして、図7における行列成分121に、これへのポインタを書き込むことによりなす。すなわち、クラスタの状態に関する構造体140には、各画素の色成分の値と個数として“1”と云う値と、そして、ルートノードのみからなる二分木150へのポインタとを書き込む。

【0144】次に、CPU11はクラスタ対の情報を保持する構造体130に全ての画素対に関しての必要な初期情報を登録する。構造体130には全ての画素対に関してその対となる各画素のデータへのポインタ131、色空間上での非類似度情報部133、および画面上での非類似度情報部134があり、構造体130におけるポインタ131には全ての画素対に関してその対となる各画素のデータへの目印となるポインタ値が登録され、そして、色空間上での非類似度情報部133には色成分の非類似度情報が登録され、また、画面上での非類似度情報部134には画面上の距離としての最小距離法の非類似度が登録される。画面上の距離としての最小距離法の非類似度としては、例えば、ユークリッド距離を求め、それを用いる。

【0145】このようにして、領域統合初期の非類似度行列は完成される。非類似度行列が完成されると、CPU11は次にブロック内の階層的な領域統合処理に入る(ステップS123-4)。

【0146】ステップS123-4におけるブロック内の階層的な領域統合処理は、まず、非類似度行列を参照し、隣接関係にあって、しかも色の類似性の高いクラスタ対から順次クラスタ統合をする。

【0147】この「隣接関係にあって、色の類似性の高いクラスタ対」の特定は例えば、この場合であると、最小距離法による非類似度が、所定値以下で、Ward法による非類似度が、最小のものを以て決める。ここで前記所定値として“1”をとれば、隣接関係は四近傍隣接になり、所定値として“ルート2(=2^{1/2})”以下とすれば八近傍隣接となる。

【0148】あるクラスタと、そのクラスタにおける四近傍隣接あるいは八近傍隣接のクラスタについてその非

類似度が最小のものとなる関係のクラスタ、すなわち、クラスタ対が特定されたならば、CPU11は次にまず、この二つのクラスタの対を一つのクラスタの情報にする。すなわち、新たなクラスタ状態情報構造体140を生成する。

【0149】クラスタ状態情報構造体140は図8(b)に示すようにラベル情報部141、画素数 n 情報部142、 μL^* 情報部143、 μa^* 情報部144、 μb^* 情報部145、領域統合履歴の二分木ルートノードへのポインタ情報部146からなり、データの生成はつぎのようにする。

【0150】まず、ラベルについては、統合される二つのクラスタのラベルの内の何れか一方を以てラベル情報とし、これをラベル情報部141に格納し、また、画素数 n 情報については上記二つのクラスタの画素数の和を求めてこれを画素数 n 情報とし、これを画素数 n 情報部142に格納し、また、統合後の色の各成分の各平均値を求め直してこれを各色成分 L^* 、 a^* 、 b^* の平均値情報格納用の情報部143、144、145に格納する。これにより、ある統合される二つのクラスタの領域統合後のデータを書き込んだクラスタ状態情報構造体140を得る。

【0151】また、領域統合履歴の二分木のルートノードへのポインタ情報部146には、図9(a)に示す如き構造の新たな二分木ノードの構造体150を生成して記録する。すなわち、二分木ノードの構造体150は、枝Pのノードへのポインタ情報部151、枝Rのノードへのポインタ情報部152、統合状態情報へのポインタ情報部153からなり、統合されるクラスタの統合履歴へのポインタを、今生成した二分木ノードの構造体150における枝Pのノードへのポインタ情報部151、枝Rのノードへのポインタ情報部152に登録する。そして、この新たに生成された二分木を、これによる統合のルートノードとして、クラスタ状態の情報構造体140のポインタ情報部146にそのポインタ情報を記録する。

【0152】また、この新たなルートノード150のデータとして、統合状態情報の構造体160を得る。統合状態情報の構造体160は、図9(b)に示す如く、ラベル情報部161、統合後の画素数 n 情報部162、 μL^* 情報部163、 μa^* 情報部164、 μb^* 情報部165、色空間上での非類似度情報部166、画面上での非類似度情報部167の各部からなり、統合後のラベル情報をラベル情報部161に、統合後の画素数 n を統合後の画素数 n 情報部162に、統合後の L^* 色成分の平均値を μL^* 情報部163に、統合後の a^* 色成分の平均値を μa^* 情報部164に、統合後の b^* 色成分の平均値を μb^* 情報部165に、統合時の色空間上での非類似度を色空間上での非類似度情報部166に、そして、統合時の画面上での非類似度を画面上での

非類似度情報部 167 に記録し、これへのポインタを 153 に記録する。

【0153】なお、初期状態すなわち 1 画素 1 クラスタのときの、二分木ノードの状態を図 10 に示す。

【0154】図 10 においては、二分木ノードの構造体 150 における枝 P のノードへのポインタ情報部 151、枝 R のノードへのポインタ情報部 152 は図 11 に示すようにヌルポインタであり、これ以上の枝を持たないこと示している。また、この二分木ノードの構造体 150 のポイントする統合状態情報 160 はその画素の状態を示している。すなわち、図 11 に示すようラベル情報部 161 はその画素のラベルの値が、また、サンプル数である画素数 n 情報部 162 には画素数として“1”が、また、色成分の平均値を格納する統合後の画素数 n 情報部 162、 μ_L 情報部 163、 μ_a 情報部 164、 μ_b 情報部 165 には、その画素の色成分値がそれぞれ格納されるが、このときの二つの非類似度情報部 166、167 は情報未設定となる。

【0155】領域統合を実施する度に、これら統合履歴が新たなルートノードの下に配置されるので、この初期状態は統合履歴二分木においてはリーフの状態となる。領域統合を次々に実施した結果、得られる統合履歴二分木を図 12 に示す。この統合履歴二分木をリストに登録し (S137)、後からこのリストを辿ることにより統合の履歴が再現できるようにする。なお、図 13 に図 12 に示す統合履歴二分木のリストノード構造体 170 を示す。

【0156】リストノード構造体 170 は次ノードへのポインタ情報を格納する次ノードへのポインタ情報部 171、前ノードへのポインタ情報を格納する前ノードへのポインタ情報部 172、データへのポインタ情報を格納するデータへのポインタ情報部 173 よりなる。

【0157】以上により、各ブロックにおける構成画素 (クラスタ) の領域統合に関する処理が終了する。統合されたクラスタに対するクラスタ状態情報構造体 40 は不要となるので、ここで破棄する。

【0158】上述したように、ステップ S123-4 において実施される統合処理は、ステップ S123-3 から S123-5 のループの処理を繰り返すことで、その時々での非類似度の最も小さいクラスタ対に対して統合してゆく。そして、領域数 (クラスタ数) が所定値 k_i になった段階で統合を終了させるが、このとき統合処理は非類似度行列の更新をしながら領域数 (クラスタ数) が所定値 k_i になるまで繰り返す。

【0159】従って、ステップ S123-5 において行われた非類似度行列の更新処理について、ここで触れておく。非類似度行列の更新処理は、最小距離法ではつぎのようにして行っている。非類似度 d なる統合前の二つのクラスタを p 、 q とし、統合されたこれらと対をなすクラスタを r とし、 p 、 q を統合して生成されたクラス

タを t としてクラスタ i 、 j 間の非類似度を d_{ij} とする時、最小距離法では、

$$d_{tr} = \min (d_{pr}, d_{qr})$$

となり、Ward 法では、

$$d_{tr} = \left\{ \frac{(n_p + n_r)}{(n_p + n_q + n_r)} \right\} d_{pr} + \left\{ \frac{(n_q + n_r)}{(n_p + n_q + n_r)} \right\} d_{qr} - \left\{ \frac{n_r}{(n_p + n_q + n_r)} \right\} d_{pq}$$

と再帰的に算出される。

【0160】ここで、 p あるいは q のいずれかと対をなすクラスタ情報を、非類似度行列から抹消し、行列成分としてはヌルポインタを代入して、登録抹消されたこと示す。つぎに、残されたクラスタ p または q に隣接する対のクラスタについて、そのクラスタ対における情報をそれらに該当するクラスタ対情報構造体 130 全てについて、その非類似度を更新する (色空間上での P 、 Q の非類似度情報部 133、画面上での P 、 Q の非類似度情報部 134 の内容更新)。

【0161】最後に、統合された二つのクラスタの対に関する情報を非類似度行列上から抹消して、ブロック単位での第 1 段階の領域統合処理が完了する。

【0162】各ブロックそれぞれについて、先に定められた暫定的な領域数 (クラスタ数) に達するまでこの処理を繰り返す。そして領域統合処理を行うことによって先の暫定的な領域数に到達した時 (その領域数に領域統合が成された時に)、各クラスタの有する統合履歴を保存する。これは例えば、各統合履歴のルートノードへのポインタをスタックする (統合履歴リスト) 等して保存する。また、必要があれば、非類似度行列を残存するクラスタのサイズに縮退させてこの非類似度行列を保管する。

【0163】このようにして、全てのブロックそれぞれについて暫定的な領域統合を実行し、その統合履歴を保管する (図 3 (b) の S123-6)。以上で、第 1 段階の領域統合処理が終わる。

【0164】[第 2 段階の領域統合処理] 第 1 段階の領域統合処理が終わると次に第 2 段階の領域統合処理に入る。

【0165】第 2 段階の領域統合処理は、第 1 段階の領域統合処理であるブロック毎の階層的領域統合によって得られた領域を初期クラスタとして用い、これら各初期クラスタを取り纏めて得た画像全体について領域統合を行う処理である (図 2 の S130、S140 の処理)。この段階では領域統合の履歴をその時々々のクラスタの色成分情報と共に保存する。そして、統合の履歴を辿り、所望の色成分の情報を呈するクラスタを知ってこれを抽出することにより、画像の所望の領域部分を分割することができるようにする。

【0166】この第 2 段階の領域統合処理においては、まず前処理としてステップ S130 の処理を実行し、各ブロック毎の領域統合の劣化を、画像全体のそれになる

ような非類似度行列を生成する。すなわち、任意のブロックにおける任意の領域とその他の任意のブロックの任意の領域との関係による非類似度行列を生成する。

【0167】第1段階の領域統合処理における各ブロック毎の処理で、縮退して非類似度行列を保管してあれば、ブロック内の領域間の関係は保持される。しかし、統合履歴のみの場合は領域間の関係は保持されていないので、この場合はここでもう一度非類似度行列を再構成せねばならない。

【0168】ここで非類似度行列を再構成する場合、最小距離法による非類似度は、隣接関係のための尺度であり、隣接関係を四近傍（四辺に接する4つ）または八近傍（四辺に接する4つと、対角位置に接する4つの計8）に限定しているのであれば、この段階では全ての領域（クラスタの対）について非類似度を求める必要はない。以下、統合履歴のみから、非類似度行列を再構成する一例を図4のフローチャートを参照して説明する。

【0169】非類似度行列を再構成するには、まず初めに、ラベル画像生成処理を行う（S131）。これはまず、ワークメモリ14に処理対象の画像と同サイズの記憶領域をラベル画像用として用意し、これにラベル画像を次のように書き込んで生成することから始める。

【0170】すなわち、ラベル画像は保管してある統合履歴を参照し（この場合はリストとして参照する）、各統合履歴のルートノードから二分木を辿り、葉（リーフ）の位置に到達したならば、その葉が指す統合初期の領域の情報（すなわち、画素の情報）からラベルを読み出し、その画素のライン（行）またはカラム（列）の座標値を算出すると共に、先に確保したワークメモリ14上のラベル画像用の記憶領域の、その座標値に対応する所にその統合履歴のルートノードのラベルの値を書き込むことで行う。

【0171】このような処理を全ての統合履歴の二分木について実行すれば、ラベル画像が得られ、このラベル画像があれば、前記第1段階での領域統合結果の画面分割の状況が、その領域のラベルの値によって参照できる状態になる。

【0172】ラベル画像が生成されたならば、次に初期非類似度行列を生成する（S132）。そして、これを用い、各領域について、上記のラベル画像上で輪郭追跡を行う（S135）。この輪郭追跡の段階で、それと相対して接した領域はこの領域に隣接する領域であり、輪郭追跡の仕方を四近傍ですか八近傍ですかにより、領域の隣接関係の限定を満たすことができる。

【0173】これを実現したものを図14に示す。図14はブロックI、IIの領域統合の様子を示している。領域Aの統合履歴から得られるラベルから、領域内の1点A1が得られる。この座標から左へ移行してブロック境界A3に達する。領域Aと他の領域との最後の境界はA2であり、これを領域Aの外側の境界上の1点として、

これより輪郭追跡を開始する。この追跡によって領域Aに接する領域が、b, d, e, a, cであることが判る。ブロック境界に接する領域eについてみれば、領域eに接する領域はb, i, k, l, m, g, n, a, Aとなる。このようにしてラベル画像上で輪郭追跡を行うことができる。

【0174】具体的に説明する。まず、初期非類似度行列としてデータ未登録状態の非類似度行列を生成し（S132）、次にもう一度、スタック（統合履歴リスト）をリストとして参照する（S134）。各領域統合履歴のルートノードのラベルはその領域内の一点となる。ラベル画像上のこの点から、上、下、右または左のいずれかに移動してゆき、領域の外側の境界を決定する。これは領域がトラス状になっている可能性があるためである。また、領域の外側の境界であることの判別は、この点のラベル画像上の移動で、その領域の属するブロックの境界に到達するまでに、最後に該領域から他の領域に移った点であるとする。

【0175】当該領域であるか否かは、ラベル画像のその時々での画像の値（ラベル値）を参照し、それが、当該領域のものと一致するか否かで識別し得る。この点が定まったならば、この点を起点として輪郭追跡を開始する（S135）。

【0176】輪郭追跡は、追跡点の右または左に当該領域外の点があるとの条件に従えば実行できる。そして、追跡点が追跡開始点と一致したところでこの追跡を終了し、次の統合履歴による処理に移る。

【0177】この輪郭追跡の際に、当該領域に接する（移動点に対して、当該領域と反対方向にある）領域の画素のラベル値を参照し、これを当該領域に接する領域とする。移動時に、直前に参照した、隣接領域のラベルの値を記憶しておけば、現隣接領域のラベル値とそれが変わった時にのみ、非類似度行列の書き込みを実行するだけで良い。

【0178】非類似度行列のインデクスは、統合履歴の登録の順番とする。

【0179】ここで新たな隣接領域が見つかった場合、統合履歴のリストから、隣接領域と同じラベルを持つルートノードを検出する。そして、当該領域とこの隣接領域に関してクラスタ対の情報構造体を生成し、この二つの領域の非類似度行列のインデクス位置にこのクラスタ対情報構造体へのポインタを登録する（図4のステップS136）。また、二つの領域の統合履歴のルートノードのクラスタ情報構造体からWard法における非類似度を算出する。また、この時の最小距離法による非類似度として、実際の距離ではなく（隣接していれば、非類似度は“1”または“ルート2（2の平方根）”になる）でも良く、ここでは、単に隣接していることを意味する“0”をセットする。

【0180】さらに、このクラスタ対の情報構造体が示

すクラスタ状態情報構造体へのポインタの欄に、統合履歴のルートノードのクラスタ情報の構造体へのポインタの値をそれぞれセットする。

【0181】二つ目の領域以降は、すでに登録された隣接関係を検出する可能性がある。また、一つ目の領域でも輪郭追跡が完了するまでに一度隣接が終わった領域に2度以上隣接することも起こり得る。この場合は、非類似度の当該二つの領域のインデックスが参照したとき、それが登録済みであるか否かを見て、未登録である場合にのみ、登録処理を実行するようにする（S136）。

【0182】以上のようにして、領域統合履歴のリストの全ての履歴について輪郭追跡による隣接領域の検出を行い、それが終了したならば（S133）、非類似度行列をそのインデックスを順次あたってゆくことにより、未登録のインデックスの対を検出する（S137）。

【0183】ステップS137の処理においては未登録のインデックスの対が、検出されたならば、インデックスに基づき、統合履歴リストを検索し、隣接領域の時と同じ要領でクラスタ対情報構造体を生成し、登録する。このとき、Ward法での非類似度は、隣接領域間で行ったと同様にして算出する。最小距離法の非類似度は隣接していないと云うことを示すために“1”とする。また、これらクラスタ対をなす各クラスタの状態情報への参照は、隣接領域間でしたのと同様になる。

【0184】CPU11による以上の操作により、第2段階の領域統合処理に用いるための非類似度行列が完成する。また、この時点でラベル画像は不必要となるので、ワークメモリ14上から消去する。

【0185】以上でステップS130の処理が終り、次にステップS140の処理に入る。ステップS140の処理は領域統合である。ここで行う領域統合処理における階層的領域統合は、基本的には第1の段階のそれと全く同じであるが、この第2段階での領域統合処理では、領域が最終的に一つの領域に統合されるまで行い、領域統合を完了させる。

【0186】この第2段階での領域統合処理でも、CPU11は第1段階での領域統合処理と同様にやはり、統合履歴を生成し続け、これを最終的な出力とする。そして、CPU11は当該生成した統合履歴を外部記憶装置17に保存する。

【0187】統合履歴の二分木の取り扱いに関しては先に先行技術で説明したものと同様であるが、階層下位においては、ブロック形状に即した領域が検出されることがあるのでその点を了解して、統合履歴を扱う必要がある。

【0188】以上の実施例は、非類似度行列による階層的領域統合について述べたが、非類似度行列の代わりにグラフ構造を用いても同様な処理結果が得られる。

【0189】また、本発明方法は必ずしも、1画面の画面全体を対象にブロックに分割する必要はなく、何らか

の前処理、例えばサンプリングピッチまたは解像度の粗い画像（ラフ画像）を利用して、一度、階層的領域統合しておいてから、抽出対象領域と背景その他の部分を決定しておき、この境界部分を記録し、今度は解像度の高いまたはサンプリングピッチの小さな画像（フル画像）のこの部分に限定して、本方法を適用することにより、領域抽出を実現すると云った手法を採用することができる。

【0190】ラフ画像の境界をフル画像に投影すれば、当然境界線は間欠的となるか、多角形近似されたものとなる。しかし、これに適当な幅を持たせたトラスにするか、矩形ブロックでこの多角形を包含するように囲み込んでゆき、これらをブロックとして扱い、本発明による階層的領域統合を行い、統合履歴を得れば、この得られた統合履歴を用いてフル画像における輪郭抽出を実施することが可能となる。

【0191】本発明の実施例を取り纏めておく。本発明の基本的な考え方は、与えられた画像に対して第1段階での領域統合処理を実施し、クラスタ数を少なくした後、これを初期クラスタとして第2段階での領域統合処理を実施する点にある。その際、隣接関係があり、色空間上の非類似度最小のクラスタから優先的に領域統合してゆくが、この時の履歴管理をする。この履歴管理には例えば、二分木で管理するという手法を採用する。

【0192】先行技術に対する本発明の異なる点は、第1段階の領域統合処理として、まず、画像を適当な計算機記憶資源に十分対応可能な大きさのブロックに分割し、このブロック内で、ある程度の階層的な領域統合を行うことによって、これら統合領域の個数を階層的クラスタ分析可能な程度に減少させ、次に第2段階の領域統合処理としてこれらの領域を再度、初期クラスタとして、領域統合を実行し、領域統合を完了させるようにした点である。

【0193】すなわち、非類似度列を用いるにせよ、グラフ構造を用いるにせよ、階層的クラスタ分析法により、画像の画面上の対比により相互に分化成立する階層的領域の抽出は、計算機上大量の記憶資源を必要とする。そこで、本発明では、第1段階の領域統合処理として、まず、画像を適当な計算機記憶資源に十分対応可能な大きさのブロックに分割し、このブロック内で、ある程度の階層的な領域統合を行うことによって、これら統合領域の個数を階層的クラスタ分析可能な程度に減少させる。そして、第1段階の統合処理により上記各ブロックでは領域統合されてデータ数（領域数）が少なくなっており、これら各ブロックを取り纏めて1画面として扱い、第2段階の領域統合処理を行うようにすることにより、第2段階の領域統合処理においても、元の画像は上記の適当な計算機記憶資源に十分対応可能な大きさで取り扱うことができるようになる。

【0194】第2段階の領域統合処理では、上記領域統

合後の領域を初期クラスタとして、再度、領域統合を実行し、領域統合を完了させる。

【0195】当然のことながら、精度上の問題等、必要があれば、この二段階の領域統合処理をさらに三段、四段と増やすようにしても構わない。また、初期時でのブロックの設定の仕方としては、最も、単純には適当なライン・カラム各方向適当なオフセットで矩形に分割してゆく方法があり、これを適用する。また、必要があれば、これら矩形のサイズは必ずしも一定にする必要はない。また、形状も矩形に限られるのではなく、不定形であっても構わない。

【0196】そして、不定形とした場合、その不定形ブロックは、1つだけであって、階層化統合を1段階にしても構わない。

【0197】領域の隣接関係を非類似度行列またはグラフ構造で管理し、領域の隣接関係の構造を生成しようとする場合は、ブロックを構成する画素の属性（主に座標値）が分かっているだけであり、任意の画素対または隣接画素の対を求めることは容易である。尚、上述した“非類似度行列で隣接関係を管理し、領域の階層的統合を実現する”とは、領域間の隣接関係を画面上での最小距離法（または、その他）で、領域間の色の類似性または対比の度合いをWard法（または、その他）で求め、これらを非類似度行列で管理すると共に、非類似度最小のものを領域統合することである。

【0198】そして、非類似度行列を用いることのメリットは、種々の組み合わせの方法を全て適用することが可能であると言う点である。非類似度行列を用いて、階層的領域統合を行う場合、例えば、以下の条件画面上で、最小距離としての非類似度が、所定値以下であり、色空間上での色の類似性がWard尺度において、所定値以下または最小値であると言う2つの条件を同時に満たすクラスタ対の中から、適宜選択されるものとする等が考えられる。

【0199】実施例では非類似度または階層的領域抽出の実行単位は、1枚の画像をブロックに区分し、階層的領域抽出を2段階で実行するものとした。この場合、まず、第1段階として、ブロック分割を行い、これらに、個々のブロックに対して統合履歴を作成して行く。領域統合は各ブロックとも、基本的には、ある程度の統合で領域統合を停止・保留する。階層的領域抽出の手法は、グラフによる階層的領域抽出は先行技術と同様であり、非類似度行列による階層的領域抽出の場合は上述の手法で実行される。また、統合履歴の生成に関してはこの段階では、先行技術と同様に二分木で記録し、二分木を辿ることで画像の明度、色相、彩度、座標等の各要素を総合的にとらえた所望の領域抽出を可能にする管理ができるようにして、画像を「ライト、中間、シャドウ」の各領域に分割することができるようにする。

【0200】第1段階での領域統合は、各ブロック内で

の統合が完了するまで実行するのではなく、所定の条件が満たされたところで停止・保留し、他のブロックの処理を同様に継続してゆく。全てのブロックについてこの段階の処理が終了した時点で、画像全体に関して、各ブロックのクラスタの関係を整理し統合し直す。

【0201】各ブロックの第1段階での領域統合の停止・保留の条件とは、基本的には、画像全体として見たときの全クラスタの対比の形成状態や、これらの関係がほぼ一定範囲内に収まり、それと同時に、全体でのクラスタの個数が、非類似度行列で、あるいはグラフで管理可能な個数になっていること、すなわち、計算機の記憶資源容量内にデータが収まった状態であるものとする。

【0202】この停止条件として考えられるものとしては、(a) ブロック内のクラスタが一定数になるか、またはそれ以下となった時、(b) ブロック内の各クラスタの色空間上での非類似度がある一定以上に達した時等である。しかし、この条件に達せず、ブロックが一つのクラスタに統合されたときはこれを以て停止とする。

【0203】各ブロックの第1段階の領域統合処理が終了した段階で、総クラスタ数がある一定値以上であれば、最も対比状態の小さいブロックからもう1回の領域統合を実行し、総クラスタ数が所定値になるまで、一領域づつのクラスタ数削減を実行する。

【0204】各ブロックの統計的パラメータ、例えば、分散値等から、各ブロックの許容クラスタ数を定め、第1段階の領域統合をこのクラスタの個数を以て停止保留する。

【0205】以上、第1の段階の領域統合処理が終了したら、第2段階の領域統合処理に入る。ここでは画像全体の領域構造への再編成のために、まず全画像のクラスタに一意のラベル付けをし、これらクラスタの全画像画面上における対の関係を求める。

【0206】これら対の関係を管理する方法としては、非類似度行列を生成する方法や、グラフ構造として管理する方法等が適用でき、その何れであっても、原理的には構わない。いずれの方法にしても、任意のブロックの任意の領域に対して、他のブロックの任意の領域との間の関係を求めることとなる。

【0207】この前処理として各領域の境界部の情報を求め、これら領域間の画面上における距離を求め直すことを考える。

【0208】その一つとして、全ての領域については輪郭追跡を行い、その境界の座標値を記録する。但し、領域とその画面上の位置を対応づけるには、統合過程を表す二分木のルートノードのラベルからその領域の内一つの画素の座標値が得られるものとする。

【0209】すなわち、1画素1クラスタを初期状態としたとき、各画素にはその座標値から一意に対応するようなラベルを割り振っておくものとする（先行技術参照）。任意の領域間の最小距離は、これら領域の境界座

標値の集合間の任意の要素間の距離のうち、最小になるものを以て定める。ここで、グラフ構造によるものは、隣接する領域、すなわち、距離ゼロのものだけに着目するので、任意のブロックの任意の領域について輪郭追跡を行い、追跡点がブロック境界上に達したとき、それに相対する境界のブロックのその座標値上にある領域とそのラベルを参照してブロック間の隣接関係を求める。平均値法・重心法・Ward法等は、新たに発生した領域対に関しては、各領域の非類似度およびそのクラスタを構成するサンプル数からこれらの非類似度を再帰的に算出することができる。

【0210】以上の手順により、再度、グラフまたは非類似度行列を構成して、再度、領域統合処理を実行する。2段階で階層的領域統合が完了するようにしたのであれば、この第2段階での領域統合処理においてクラスタが一つに統合された段階で、階層的領域統合を終了する。

【0211】そして、領域統合にあたっては、統合した領域がその時々でどのようになったかを後から辿ることができるようにするために、二分木による履歴管理をしており、その時々で統合されたクラスタの明度、色相、彩度、座標等の各要素の情報を含めて履歴管理するようにしているので、この履歴管理の情報により、画像の明度、色相、彩度、座標等の各要素を総合的にとらえた所望の領域抽出を可能にする。

【0212】以上のように本発明は、処理対象の画像を適宜な単位に分割あるいはブロック化し、この分割あるいはブロック化したものを単位に、第1段階での領域統合を行い、この第1段階での領域統合が済んだ後、この処理済みの画像を初期クラスタとして用い、これら初期クラスタを取り纏めて元の1画面分の画像として扱い、第2段階での領域統合を行うと共に、統合の履歴をその時々でクラスタの色成分情報と共に保存しておき、そして、統合の履歴を辿り、所望の色成分の情報を呈するクラスタを知ってこれを抽出することにより、画像の所望の領域部分を分割することができるようにするものである。

【0213】本発明では処理対象の画像を分割あるいはブロック化して領域統合処理を行うことで、領域統合処理毎の一度に取り扱う画素・クラスタ・領域の個数を制限するようにし、これによって記憶資源の小さな電子計算機においても演算処理が可能となるように改善した。

【0214】すなわち、自然画像における階層的領域抽出の実行（自然画像における画面上の色の分布の大局的対比の階層的関係・構造を抽出する処理）を、全画面に対して一回で終了させようとする、グラフ構造により、隣接する領域関係の管理だけをしても、相当量の記憶資源が必要となる。例えば、四近傍隣接の管理だけをすると、隣接関係の組み合わせは、縦横（ $h \times w$ ）画素の画像においては

$$2hw - (w + h)$$

となる。これは、全ての画素の組み合わせ（非類似度行列を用いる場合）を取り扱えば、

$$hwC2 = hw(hw - 1) / 2$$

となり、明らかに前者は、発生し管理しなければならない画素対の最大が、総画素数に比例し、後者はその最大数が総画素数の二乗に比例する。それでも、グラフ構造を用いて実現しようとする場合、一つのクラスタあたり200から300Byteの記憶容量を必要とし、 512×512 程度の画像においても100MByte近頃の記憶容量が必要となる。

【0215】膨大なデータを電子計算機で処理する場合、処理速度を重視するならば、データを電子計算機の主記憶（メインメモリ）におくことが望ましい。そして、主記憶として、上述のような記憶容量のメモリを確保することは、実現性と云う視点でとらえれば、全く不可能な数字ではないが、その一方で現時点における記憶資源のコストが高価である云う現実があることから、これ程の記憶容量を主記憶として用意するには経済的に制約が大きい。また、主記憶の代わりに仮想記憶を用いることも可能であるが、外部記憶とのデータの授受は膨大なオーバーヘッドとなり、処理時間が急激に増大することとなるから、実用性の面で問題が大きい。

【0216】また、グラフ構造により近傍隣接する画素の対のみを管理するようにする場合、全ての画素の対の非類似度の情報はなため、それで、再帰的に非類似度を算出する方法（組み合わせ的方法）の適用範囲が限定される。

【0217】そこで、本発明では処理対象の画像を適宜な単位に分割あるいはブロック化し、この分割あるいはブロック化したものを単位に、第1段階での領域統合を行い、この第1段階での領域統合が済んだ後、この処理済みの画像を取り纏めて元の1画面分の画像として扱い、第2段階での領域統合を行って、目的の領域の抽出を可能にする。本発明では処理対象の画像を分割あるいはブロック化して領域統合処理を行うことで、領域統合処理毎の一度に取り扱う画素・クラスタ・領域の個数を限定することができようとし、これによって記憶資源の小さな電子計算機においても演算処理が可能となるように改善した。

【0218】例えば、画面を 32×32 （1024）画素の単位となるブロックに分割するとする。非類似度行列として発生する画素の対の組み合わせは、 523776 個となり、画素対1あたりのデータ量が仮に50Byte（非類似度行列のほうがデータ構造はグラフ構造に比して簡単であるので、処理単位あたりに要する記憶資源は小さい）とすれば、この行列を主記憶上に構成するのに必要とされる記憶容量は、約25MByteとなり、標準的な計算機の記憶資源の大きさととなる。また、非類似度行列を用いる場合は、処理単位、即ち画素数が少なく

なれば、二乗に比例して画素対の組み合わせが減少する。(この条件で、グラフ構造を用いた場合は、初期の画素対の個数は1984個、記憶資源の必要量は、約0.55MByteとなる)。

【0219】各ブロックでの領域統合は、完了させるのではなく、非類似度またはクラスタ数等が適当な条件に達した所で停止保留し、これらの統合履歴を保存し、この処理を他の全てのブロックに対して実行する。全てのブロックに対してこの処理が終了した段階で、任意のブロックの任意の領域を他のブロックの他の領域との関係をしらべ、全画像については、再度、グラフ構造または非類似度行列を構成し、階層的領域統合の処理を再度実行して、領域統合を完成させる。

【0220】各ブロックでの領域統合の停止は、2回目の領域統合を行うに際して、画像全体でグラフ構造または非類似度行列が組める程度の領域数に減少していること、また、各クラスタ対の非類似度の分布ができるだけ均一となっているようにすることがポイントとなる。

【0221】また、画像を正方形のブロックの単位で分割した例での説明を行ったが、画像の画面の局所局所に適合してブロックは任意の矩形としても構わなければ、不定形であっても構わない。また、ブロックが一つで一度に階層的領域抽出が実現できる場合も本法の特種形として扱うものとする。

【0222】本発明により、画像の階層的領域抽出が、比較的主記憶の小さな計算機においても実現可能となる他、また、処理速度に関しては、非類似度行列を用いる場合、ブロック単位を小さくすることにより処理効率が飛躍的に改善されるようになる。

【0223】また、1画面をブロックに分割することで特定の画像部分に限定して、階層的領域統合を実行することが可能となる。また、これにより、生成される領域統合履歴の記憶容量を少なくできるようになり、処理によっては不必要な領域統合履歴を省くことも可能となる。

【0224】また、非類似度行列に対応できるようになったことで、グラフ構造では大幅に制約を受けていた組み合わせの方法による非類似度の算出方法に制約がなくなった。

【0225】なお、本発明は上述した実施例に限定することなく、種々変形して実施し得る。

【0226】

【発明の効果】以上、詳述したように本発明によれば、自然画像における画面上の色の分布の大局的対比の階層的関係・構造を抽出する処理を、比較的主記憶の小さな計算機においても実現可能となり、また、処理速度に関しては、非類似度行列を用いる場合、ブロック単位を小さくすることにより処理効率が飛躍的に改善される等の効果が得られる。

【図面の簡単な説明】

【図1】本発明の実施例を説明するための図であって、本発明の一実施例に係る画像処理装置の構成を示すブロック図である。

【図2】本発明の実施例を説明するための図であって、本発明の一実施例に係る全体的な処理手順の概要を説明するためのフローチャートである。

【図3】本発明の実施例を説明するための図であって、本発明の一実施例に係る領域統合処理の詳細を説明するためのフローチャートである。

【図4】本発明の実施例を説明するための図であって、本発明の一実施例に係る第2段階での領域統合処理の詳細を説明するためのフローチャートである。

【図5】本発明の実施例を説明するための図であって、1枚の画像を一定サイズの矩形でブロックに分割して分割画像化した例を示す図。

【図6】本発明の実施例を説明するための図であって、各ブロック(各分割画像)の詳細を示した図である。

【図7】本発明の実施例を説明するための図であって、本発明の一実施例に係る非類似度行列の例を示す図である。

【図8】本発明の実施例を説明するための図であって、本発明の一実施例に係る情報構造体のうちの画素対の情報を有する情報構造体130および各クラスタの状態の情報を保持する構造体140を説明するための図である。

【図9】本発明の実施例を説明するための図であって、本発明の一実施例に係る二分木ノードの構造体150および統合状態情報の構造体160を説明するための図である。

【図10】本発明の実施例を説明するための図であって、本発明の一実施例における初期状態(1画素1クラスタ)のときの、二分木ノードの状態を示す図である。

【図11】本発明の実施例を説明するための図であって、本発明の一実施例における初期状態時での二分木ノードの構造体150の内容例を示す図である。

【図12】本発明の実施例を説明するための図であって、本発明の一実施例に係る領域統合を次々に実施した結果、得られる統合履歴二分木を示す図である。

【図13】本発明の実施例を説明するための図であって、本発明の一実施例に係る統合履歴二分木のリストノード構造体170の例を示す図である。

【図14】本発明の実施例を説明するための図であって、一例としてのブロックI、IIの領域統合の様子を示した図である。

【図15】先行技術の説明をするための図であって、先行技術の一実施例に係る画像データ、グラフ構造のデータ等を表わす図である。

【図16】先行技術の説明をするための図であって、先行技術の一実施例に係る対比強度尺度値の計算式および $L^* a^* b^*$ 色空間を表わす図である。

【図 17】先行技術の説明をするための図であって、先行技術の一実施例に係る画像処理装置の領域統合処理を表わすフローチャートである。

【図 18】先行技術の説明をするための図であって、先行技術の一実施例に係る頂点のデータ、辺のデータを表わす図である。

【図 19】先行技術の説明をするための図であって、先行技術の一実施例に係る頂点のデータ、辺のデータ、リストデータを表わす図である。

【図 20】先行技術の説明をするための図であって、先行技術の一実施例に係る頂点のデータ、ノードデータを表わす図である。

【図 21】先行技術の説明をするための図であって、先行技術の一実施例に係る B-木等を表わす図である。

【図 22】先行技術の説明をするための図であって、先行技術の一実施例に係る画像データを表わす図である。

【図 23】先行技術の説明をするための図であって、先行技術の一実施例に係る領域分割された画像データを示す図である。

【図 24】先行技術の説明をするための図であって、先行技術の一実施例に係る領域統合を表わすツリーであ

る。

【図 25】先行技術の説明をするための図であって、先行技術の一実施例に係る領域分割された画像データを示す図である。

【図 26】先行技術の説明をするための図であって、先行技術の一実施例に係る領域統合過程における頂点および辺を表わすグラフである。

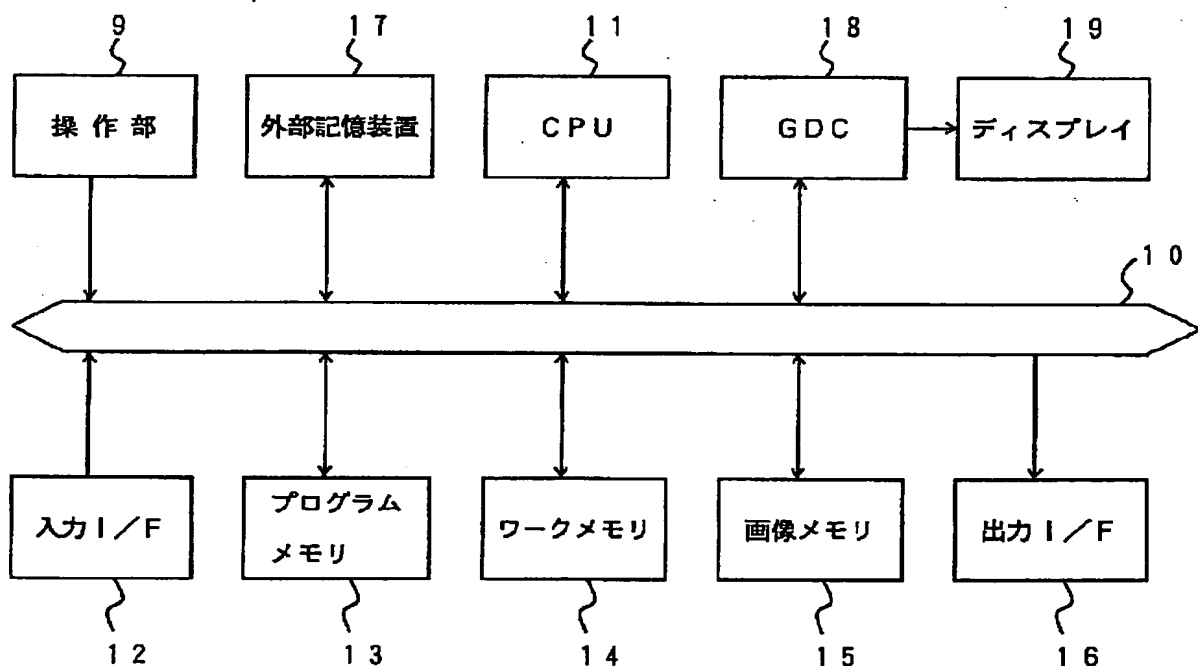
【図 27】先行技術の説明をするための図であって、先行技術の一実施例に係る領域統合過程における頂点および辺を表わすグラフである。

【図 28】先行技術の説明をするための図であって、先行技術の一実施例に係る領域統合過程における頂点および辺を表わすグラフである。

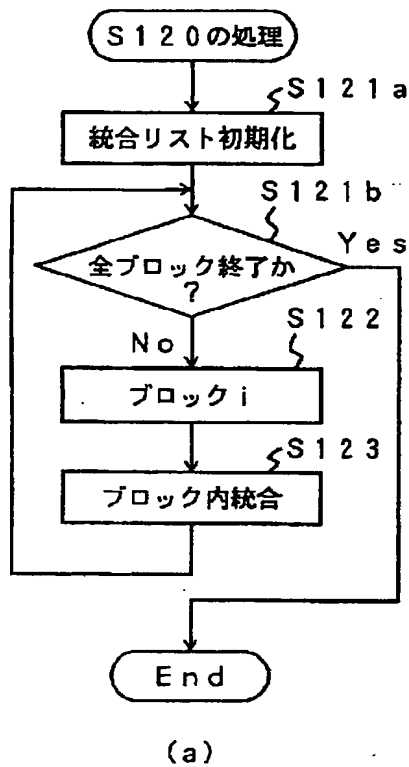
【符号の説明】

- 11 CPU (プロセッサ)
- 12 入力 I/F (インタフェース)
- 13 プログラムメモリ
- 17 外部記憶装置
- 18 GDC (グラフィック・ディスプレイ・コントローラ)
- 19 ディスプレイ

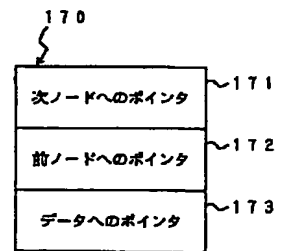
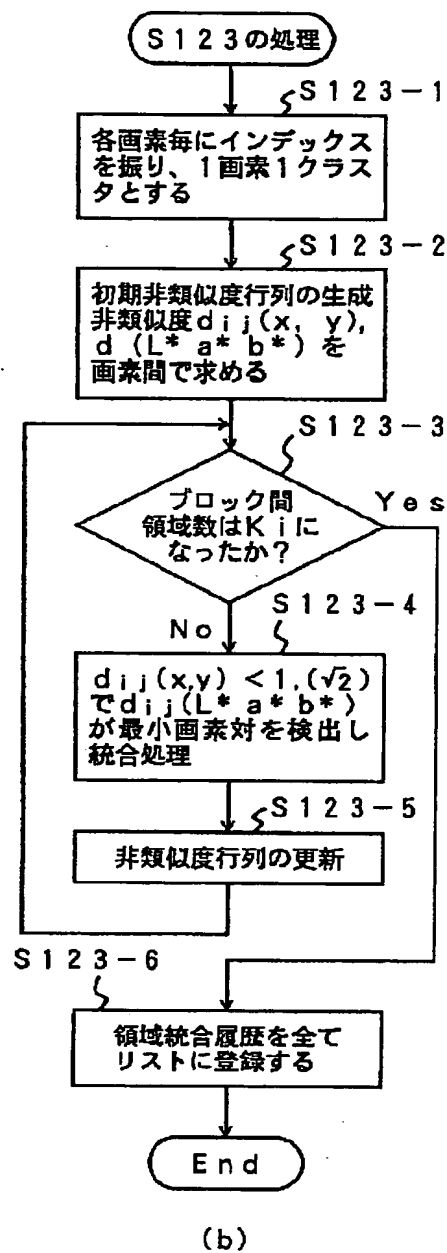
【図 1】



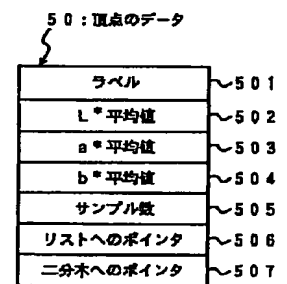
【図3】



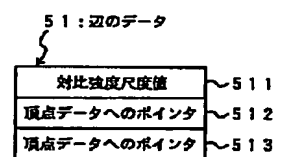
【図13】



【図18】

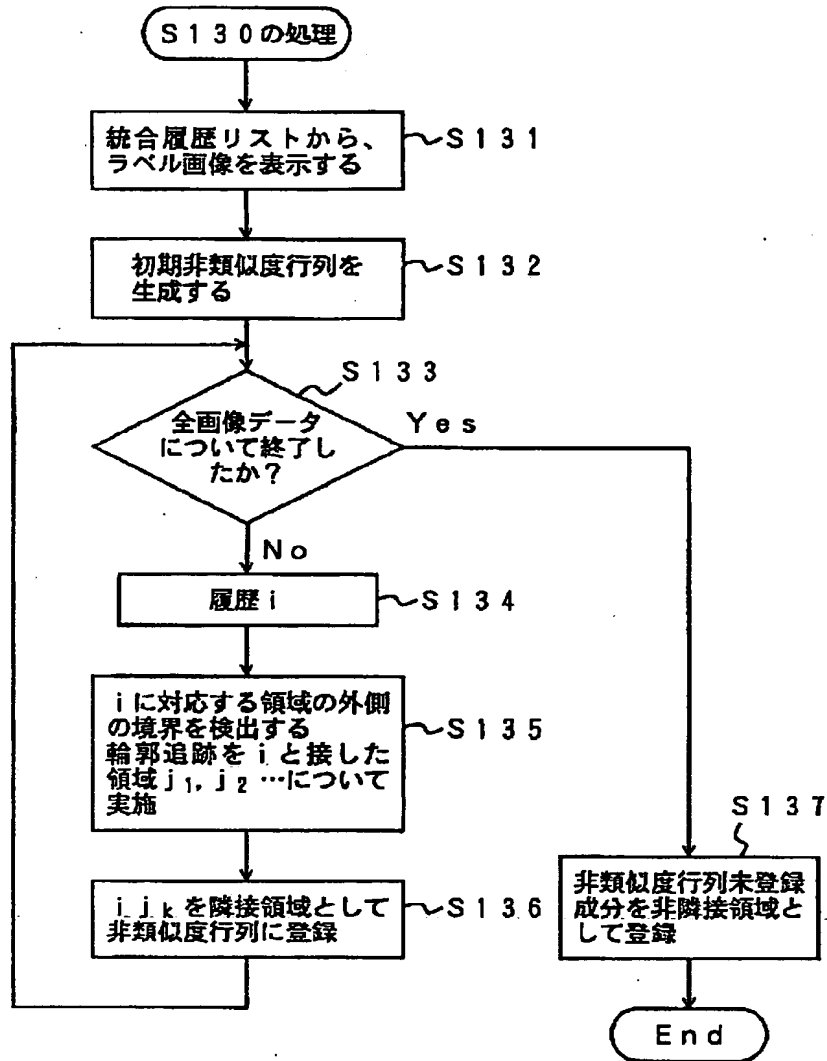


(a)

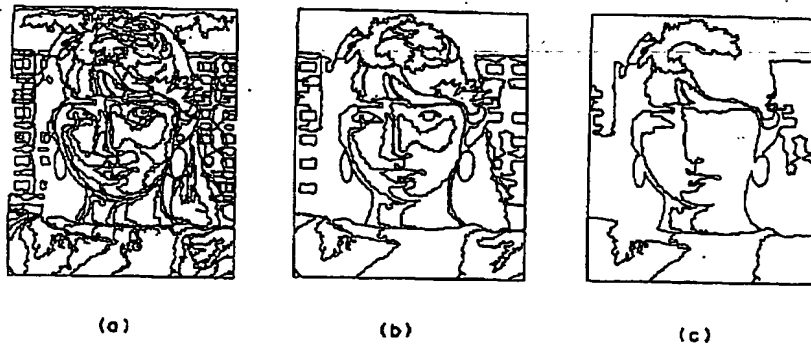


(b)

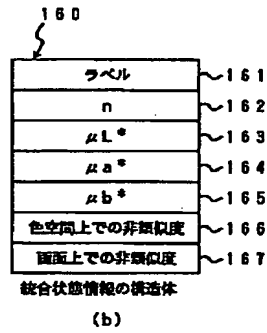
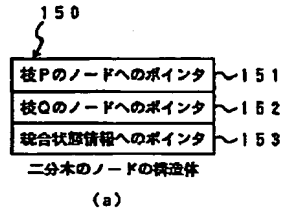
【図4】



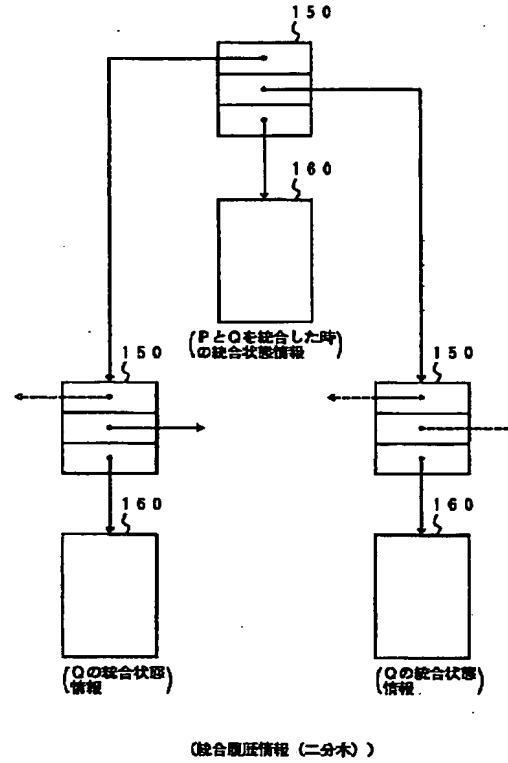
【図23】



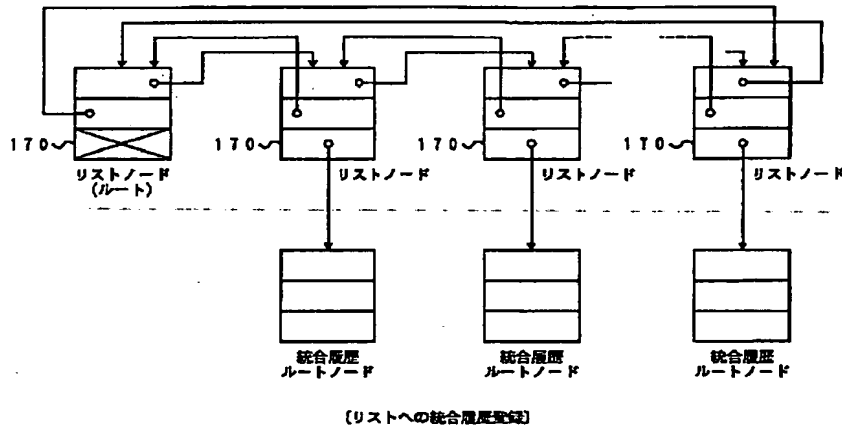
【図9】



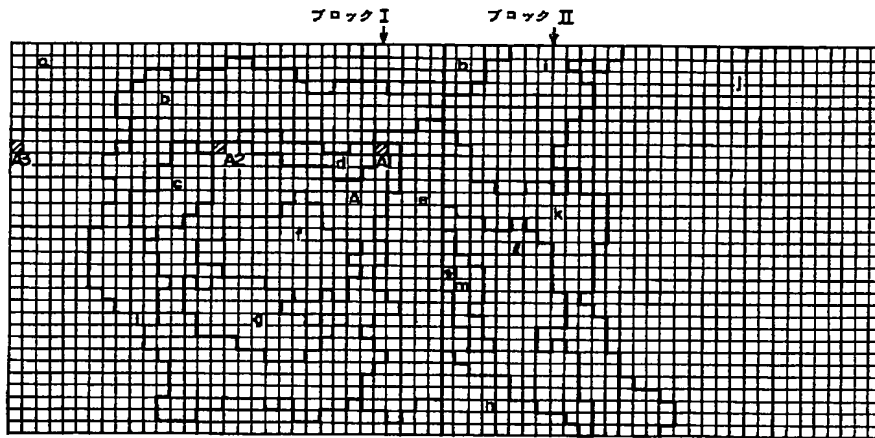
【図10】



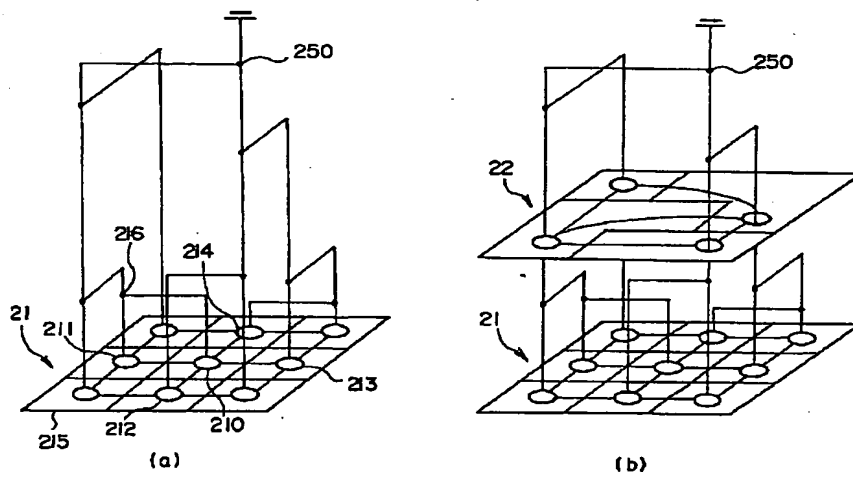
【図12】



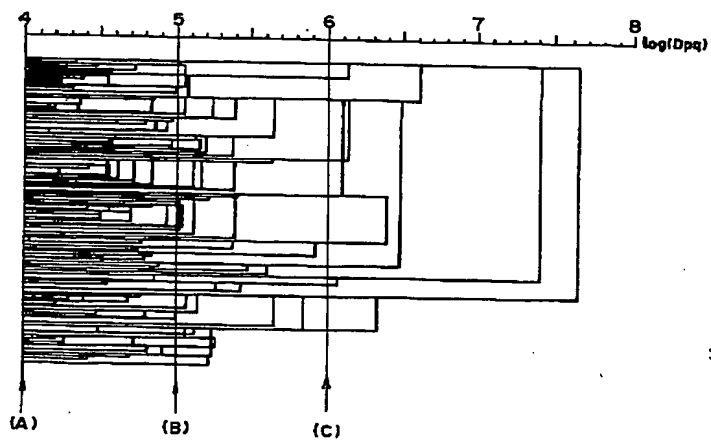
【図 14】



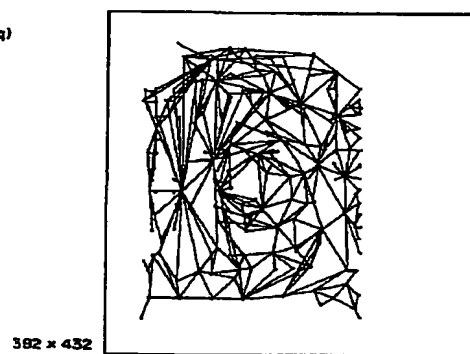
【図 15】



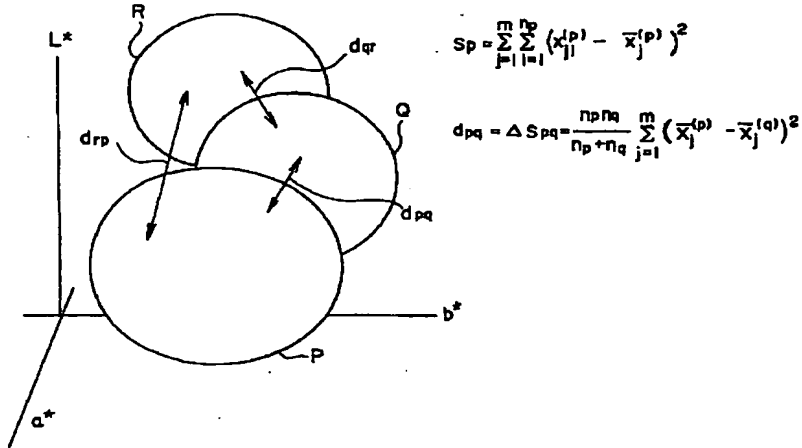
【図 24】



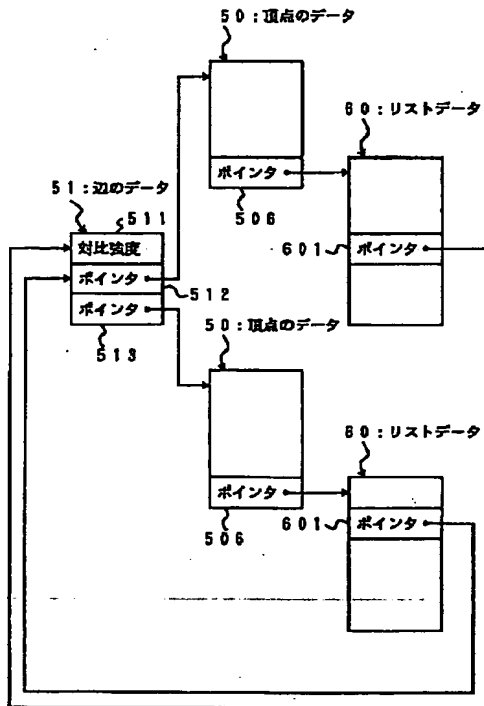
【図 26】



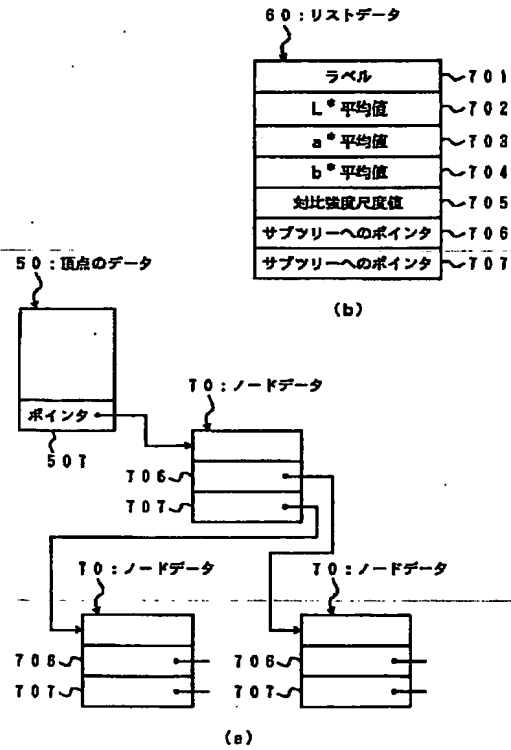
【図 16】



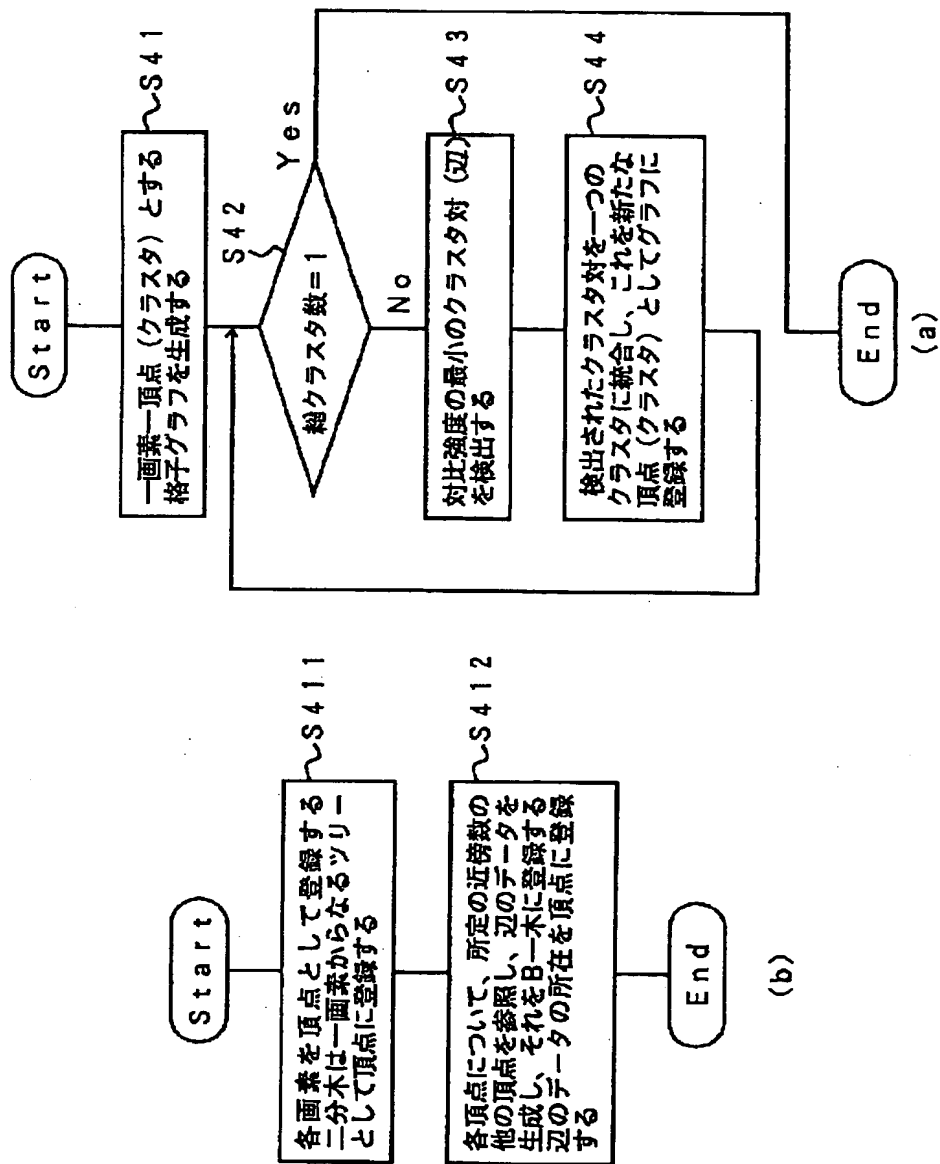
【図 19】



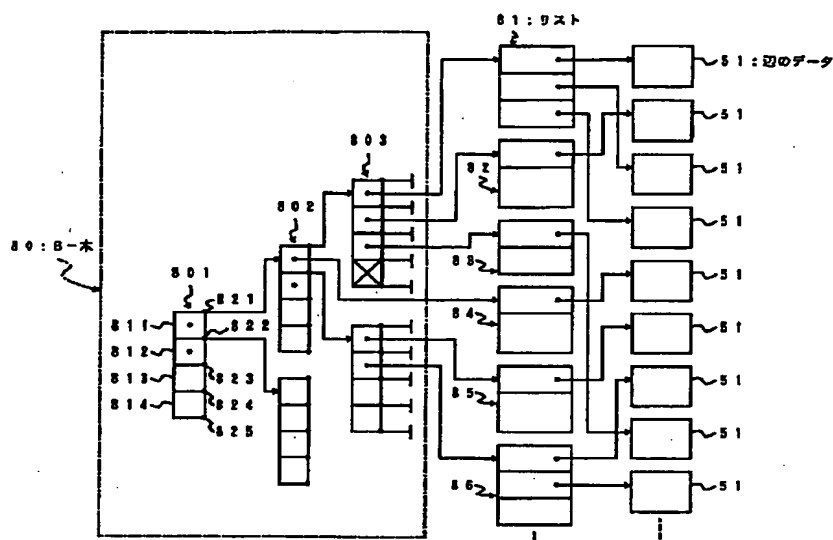
【図 20】



【図 17】



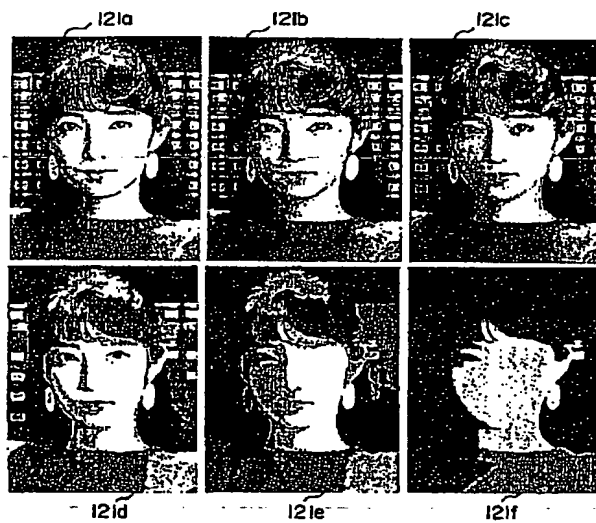
【図 21】



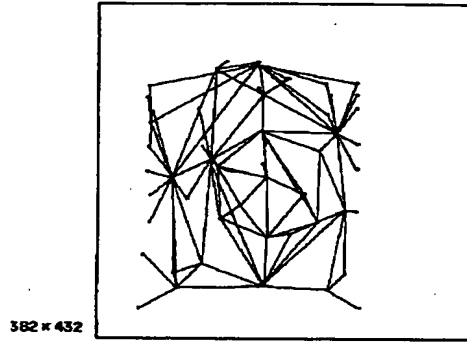
【図 22】



【図 25】



【図27】



【図28】

